Z-29 TERMINAL ROM SOURCE LISTING

Z-29 Terminal

ROM Source Listing

This document contains the source listing of the code contained in the Z-29 Terminal's ROM.

Reproduction of this code (software) in any form, except with written consent from Zenith Data Systems Corporation, is expressly prohibited.

NOTICE

Zenith Data Systems Corporation makes no express or implied warranties with regard to this software, including merchantability, fitness for any purpose, or non-infringement of patents, copyrights, or other proprietary rights of others. Zenith Data Systems Corporation shall not have any liability or responsibility to sublicensees, including end-users, for damages of any kind, including special, indirect, or consequential damages, arising out of or resulting from any program, services, or materials made available hereunder or the use or modification thereof.

The information in this publication is considered to be correct and up to date when it is first published. Zenith Data Systems Corporation reserves the right to change the product at any time without incurring any obligation to incorporate new features in products already sold. These changes may or may not be reflected in this or future editions of this documentation.

595-3053-01 Copyright 1983 Zenith Data Systems Corporation *All Rights Reserved* First Printing Printed in the United States of America



Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FE	B-83 15:30:06 PAGE 1
1 133	Z-29 Terminal Firmware Version 1.03
2 3 4	THEVERED BOTH MICH M. Dalbard Cohenago 10 1000
4 ; 5 ;	Written by R. Nick K. Boland February 10, 1983
7	Copyrisht (C) 1983 Zenith Data Systems
\$	Saint Joseph, Michigan
	Initial consultation by Tom Blog
12 ;	Initial hardware by Michael Gorbutt Hardware finished by Anthony M. Olson
13 ; 	
15 ;	Many things have been used to compact the program. Look for these situations. The most common are where subroutines
	use parts of other subroutines. Other instances are where
19 ;	a single LUMP could be used and many AUMP's used to reference the long Jump.
21 ;	
	ENABL LC NLIST TOC
23 24 25 26	NAME Z29ROM SBTTL *** Z-29 Computer Terminal Firmware Version 1.03 ***

))))))))

)

avaaa36	ожне отъестива. Г. IUMA	: 			(Alapase	
2				unenastetequi.	/alences	
		NULL	EQU			
5	0007 0008	BELL BS	EQU	00000111B	; Bell	
	0009	₽≽	EQU EQU	00001000B	; Backspage ; Horizontal tab	
7	000A	LF.	EQU		; Line feed	
8	000D	CR	EQU	00001101B	; Carriage return	
9 10	0011	XON XOFF	EQU EQU	00010001B 00010011B	; DC1 (XON) ; DC3 (XOFF)	
	0018	CON	EQU	00011000B	; Cancel	
12 13	001B	ESC	EQU	00011011B	; Escape	
14	007F	RUBOUT	EQU	01111111B	; Rubout / delete	
	•••••					
16 17		*	Const			
18	0018	MAXLINE	EQU	24	; Max number of lines is 024	
12	0019	NUMLINES	EQU	25 79	Number of lines is 125; Max number of chars/line 079	
20 21	00 4F 0050	MAXCHAR NUMCHARS	EQU EQU	79 . 80	; Max number of chars/line 079 ; Number of characters is 180	
21 22					7 Mailbei of Chai acters 13 1oo	
23 24						
		; ;	8051	processor consta	ınts	
26						
27 28	0000	BANKO BANK1	EQU	00000000B	; Select register bank 0 ; Bank 1	
29 30	0023	ITMOD	EQU	00100011B	; Bank 1 ; Init value for -TMOD-	
30	005A	ISCON	EQU	01011010B	; Init value for -SCON-	• • • • • • • • • • • • • • • • • • • •
31 32	0003 0095	IFINIT IEINIT	EQU EQU	00000011B 10010101B	; Init value for interrupt priorities	
33	0082	IEDMA	EQU	10000010B	; Init value for interrupts ; Init value for interrupts during DMA	
					······································	
	•••••					
	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		· · · · · · · · · · · · · · · · · · ·		
	•••••		• • • • • • • • • • • • • • • • • • • •	••••••		
					•	
		• • • • • • • • • • • • • • • • • • • •		***************************************		
• • • • • • • • • • • • • • • • • • • •	••••••					
•••••						
				••••••	······································	
	•••••					
					······································	
	***************************************		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		

*** Z-29 CON	MPULER TERMINAL FIRM	MWARE VERSION 1.03 ***			
			Keyboa	ard data	
2 3		; ;	Data f	from the keybo	ard other than ASCII data is in the format s = shift status, c = control status and
4 5		7	the re	est of the work	d defines the function. Exceptions to this
·····			are th	he access func	tion, power up function and ID number.
		······		OaaaaaaB	ASCII data to be transmitted
9 10	0040	; KB_SHFT	·····EQU	01000000B	; Keyboard shift mask
10	0040	KB_CNTR	EQU	00100000B	; Keyboard control mask
12					
13			EQU	10000000B	; Up arrow key
14	0080 0081	KB_UP KB_DOWN	EQU	10000000B	; Down arrow key
15 16	0081	KB_LEFT	·····ĒĠŪ····	10000010B	; Left arrow key
17	0083	KB_RIGHT	EQU	10000011B	; Right arrow key
18	0084	KB_HOME	EQU	10000100B	; Home key
19	0085	KB_ERASE	EQU	10000101B	; Erase key
20	0086	KB_HELP	EQU	10000110B	; Help key
21	0087	KB_SCROLL KB_SETUP	EQU EQU	10000111B	; Scroll key ; Setup key
22	0088 0089	KB_BREAK	EQU	10001000D	; Break key
23 24	0082 008A	KB_CAPS	EQU	10001010B	; Caps lock key
25 25	QOSB	KB_TAB	EQU	10001011B	7 Tab key
26	008C	KB_SPACE	EQU	10001100B	; Space bar
27					······································
28 29	008F	KB_POWER	EQU	10001111B	; Keyboard just powered up
30	0090	KB_O	EQU	10010000B	; Numeric kerpad O ker
31	00?1	KB1	EQU	10010001B	Numeric keypad 1 key
32	0092	KB_2	EQU EQU	10010010B 10010011B	; Numeric keypad 2 key ; Numeric keypad 3 key
33 34	9093	KB_3 KB_4	EQU	10010100B	; Numeric keyrad 4 key
3 4 35	90?5	KB_5	EQU	10010101B	; Numgric keypad 5 key
36	0096	KB_6	EQU	10010110B	; Numeric keypad 6 key
37	0097	KB_7	EQU	10010111B	; Numeric keypad 7 key
38	0098	KB_8	EQU	10011000B	Numeric keypad 8 key
3.9	0099	KB?	EQU	10011001B	; Numeric keypad 9.key
40	009A	KB_DEC KB_ENT	EQU	10011010B	; Numeric keypad decimar Point ker ; Numeric keypad enter key
41 42	QQ9B 009C	KB_DASH	EQU	10011100B	Numeric keypad dash key
43	007C	KB_COMMA	EQU	10011101B	; Numeric keypad comma key
,					
			• • • • • • • • • • • • • • • • • • • •		

))))))

)

•

)

			^		LL_L C.11
2		**************************************		s.code.and.kexs	THAT FOLIQU
			Theas	cess code from	the keyboard is very much like an escape
4		5	charac	ter. It indica	tes that the second character from the
			kembaa	ar.d. will l. be. the .	data.with.the.following.definitions
6	0005	; 			
	009F	KB_ACC		1001.111118	
9	0080	KB_F1	EQU	10000000	. Franchista III
<u>9</u>	0081	 КВ_F2	EQU	10000000B 10000001B	
11	0082	KB_F3	EQU	10000001B	; Function key #3
12	0083	KB_F4	EQU	10000011B	; Function key #4
13	0084	KB_F5	EQU	10000100B	; Function key #5
14	0085	KB_F6	EQU	10000101B	; Function key #6
15		KB_F7	EQU	10000110B	; Function key #7
16	0087	KB_F8	EQU	10000111B	; Function key #8
17 18	0088	KB_F9	EQU	10001000B	; Function key #9
	0000	1700			
<u>19</u>	0098	KB-ID	EQŲ	10011000B	; ID number 10011xxxB
2 <u>1</u>	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	
		5 5	Keyboa	ird commands	***************************************
25 26		5			
26		5	These	are commands the	at are sent from the terminal to the
27			keyboa	rd. They tell	the keyboard to enter and exit different
28		;	modes	and to perform	different functions.
29 30					
		KBC_ID	EQU	10000000B	5 Send keyboard ID number
				10000001B	* Di L-11
31	0081	KBC_BELL	EQU		3 Rina pell
32	0082	KBC_EC	EQU	10000010B	; Enable key click
32 33	0082 0083	KBC_EC KBC_DC	EQU EQU	10000010B 10000011B	; Enable key click ; Disable key click
32 33 34	0082 0083 0084	KBC_EC KBC_DC KBC_EL	EQU EQU EQU	10000010B 10000011B 10000100B	; Enable key click ; Disable key click ; Enter caps locked
32 33	0082 0083 0084 0085	KBC_EC KBC_DC KBC_EL KBC_XL	EQU EQU EQU	10000010B 10000011B 10000100B 10000101B	; Enable key click ; Disable key click ; Enter caps locked ; Exit caps locked
32 33 34 35 36	0082 0083 0084	KBC_EC KBC_DC KBC_EL KBC_XL KBC_ER	EQU EQU EQU EQU	10000010B 10000011B 10000100B 10000101B 10000110B	; Enable key click ; Disable key click ; Enter caps locked ; Exit caps locked ; Enable auto repeat mode
32 33 34 35 36 37 38	0082 0083 0084 0085 0086	KBC_EC KBC_DC KBC_EL KBC_XL KBC_ER KBC_DR	EQU EQU EQU EQU EQU EQU	10000010B 10000011B 10000100B 10000101B 10000110B 10000111B	; Enable key click ; Disable key click ; Enter caps locked ; Exit caps locked ; Enable auto repeat mode ; Disable auto repeat mode
32 33 34 35 36 37 38	0082 0083 0084 0085 0086 0087	KBC_EC KBC_DC KBC_EL KBC_EL KBC_ER KBC_DR KBC_DK	EQU EQU EQU EQU EQU EQU EQU	10000010B 10000011B 10000100B 10000101B 10000110B 10000111B	; Enable key click ; Disable key click ; Enter caps locked ; Exit caps locked ; Enable auto repeat mode ; Disable auto repeat mode ; Disable keyboard
32 33 34 35 36 37 38 39 40	0082 0083 0084 0085 0086 0087 0088 0089	KBC_EC KBC_DC KBC_EL KBC_XL KBC_ER KBC_DR	EQU EQU EQU EQU EQU EQU	10000010B 10000011B 10000100B 10000101B 10000110B 10000111B	; Enable key click ; Disable key click ; Enter caps locked ; Exit caps locked ; Enable auto repeat mode ; Disable auto repeat mode ; Disable keyboard ; Enable keyboard
32 33 34 35 36 37 38 39	0082 0083 0084 0085 0086 0087 0088 0089	KBC_EC KBC_DC KBC_EL KBC_XL KBC_ER KBC_DR KBC_DK KBC_EK	EQU EQU EQU EQU EQU EQU EQU	10000010B 10000011B 10000100B 10000110B 100001110B 10000111B 10001000	; Enable key click ; Disable key click ; Enter caps locked ; Exit caps locked ; Enable auto repeat mode ; Disable auto repeat mode ; Disable keyboard ; Enable keyboard ; Turn on offline LED
32 33 34 35 36 37 38 39 40	0082 0083 0084 0085 0086 0087 0088 0089	KBC_EC KBC_DC KBC_EL KBC_XL KBC_ER KBC_DR KBC_DK KBC_EK KBC_EK	EQU EQU EQU EQU EQU EQU EQU EQU EQU	10000010B 10000011B 10000100B 10000101B 10000111B 100001000B 10001001B	; Enable key click ; Disable key click ; Enter caps locked ; Exit caps locked ; Enable auto repeat mode ; Disable auto repeat mode ; Disable keyboard ; Enable keyboard
32 33 34 35 36 37 38 39 40	0082 0083 0084 0085 0086 0087 0088 0089	KBC_EC KBC_DC KBC_EL KBC_XL KBC_ER KBC_DR KBC_DK KBC_EK KBC_EK	EQU EQU EQU EQU EQU EQU EQU EQU EQU	10000010B 10000011B 10000100B 10000101B 10000111B 100001000B 10001001B	; Enable key click ; Disable key click ; Enter caps locked ; Exit caps locked ; Enable auto repeat mode ; Disable auto repeat mode ; Disable keyboard ; Enable keyboard ; Turn on offline LED
32 33 34 35 36 37 38 39 40	0082 0083 0084 0085 0086 0087 0088 0089	KBC_EC KBC_DC KBC_EL KBC_XL KBC_ER KBC_DR KBC_DK KBC_EK KBC_EK	EQU EQU EQU EQU EQU EQU EQU EQU EQU	10000010B 10000011B 10000100B 10000101B 10000111B 100001000B 10001001B	; Enable key click ; Disable key click ; Enter caps locked ; Exit caps locked ; Enable auto repeat mode ; Disable auto repeat mode ; Disable keyboard ; Enable keyboard ; Turn on offline LED
32 33 34 35 36 37 38 39 40	0082 0083 0084 0085 0086 0087 0088 0089	KBC_EC KBC_DC KBC_EL KBC_XL KBC_ER KBC_DR KBC_DK KBC_EK KBC_EK	EQU EQU EQU EQU EQU EQU EQU EQU EQU	10000010B 10000011B 10000100B 10000101B 10000111B 100001000B 10001001B	; Enable key click ; Disable key click ; Enter caps locked ; Exit caps locked ; Enable auto repeat mode ; Disable auto repeat mode ; Disable keyboard ; Enable keyboard ; Turn on offline LED
32 33 34 35 36 37 38 39 40	0082 0083 0084 0085 0086 0087 0088 0089	KBC_EC KBC_DC KBC_EL KBC_XL KBC_ER KBC_DR KBC_DK KBC_EK KBC_EK	EQU EQU EQU EQU EQU EQU EQU EQU EQU	10000010B 10000011B 10000100B 10000101B 10000111B 100001000B 10001001B	; Enable key click ; Disable key click ; Enter caps locked ; Exit caps locked ; Enable auto repeat mode ; Disable auto repeat mode ; Disable keyboard ; Enable keyboard ; Turn on offline LED
32 33 34 35 36 37 38 39 40	0082 0083 0084 0085 0086 0087 0088 0089	KBC_EC KBC_DC KBC_EL KBC_XL KBC_ER KBC_DR KBC_DK KBC_EK KBC_EK	EQU EQU EQU EQU EQU EQU EQU EQU EQU	10000010B 10000011B 10000100B 10000101B 10000111B 100001000B 10001001B	; Enable key click ; Disable key click ; Enter caps locked ; Exit caps locked ; Enable auto repeat mode ; Disable auto repeat mode ; Disable keyboard ; Enable keyboard ; Turn on offline LED
32 33 34 35 36 37 38 39 40	0082 0083 0084 0085 0086 0087 0088 0089	KBC_EC KBC_DC KBC_EL KBC_XL KBC_ER KBC_DR KBC_DK KBC_EK KBC_EK	EQU EQU EQU EQU EQU EQU EQU EQU EQU	10000010B 10000011B 10000100B 10000101B 10000111B 100001000B 10001001B	; Enable key click ; Disable key click ; Enter caps locked ; Exit caps locked ; Enable auto repeat mode ; Disable auto repeat mode ; Disable keyboard ; Enable keyboard ; Turn on offline LED
32 33 34 35 36 37 38 39 40	0082 0083 0084 0085 0086 0087 0088 0089	KBC_EC KBC_DC KBC_EL KBC_XL KBC_ER KBC_DR KBC_DK KBC_EK KBC_EK	EQU EQU EQU EQU EQU EQU EQU EQU EQU	10000010B 10000011B 10000100B 10000101B 10000111B 100001000B 10001001B	; Enable key click ; Disable key click ; Enter caps locked ; Exit caps locked ; Enable auto repeat mode ; Disable auto repeat mode ; Disable keyboard ; Enable keyboard ; Turn on offline LED
32 33 34 35 36 37 38 39 40	0082 0083 0084 0085 0086 0087 0088 0089	KBC_EC KBC_DC KBC_EL KBC_XL KBC_ER KBC_DR KBC_DK KBC_EK KBC_OFFLN KBC_OFFLN KBC_ONLN	EQU EQU EQU EQU EQU EQU EQU EQU	10000010B 1000010B 10000101B 10000110B 10000111B 10001000	; Enable key click ; Disable key click ; Enter caps locked ; Exit caps locked ; Enable auto repeat mode ; Disable auto repeat mode ; Disable keyboard ; Enable keyboard ; Turn on offline LED ; Turn off offline LED
32 33 34 35 36 37 38 39 40	0082 0083 0084 0085 0086 0087 0088 0089	KBC_EC KBC_DC KBC_EL KBC_XL KBC_ER KBC_DR KBC_DK KBC_EK KBC_OFFLN KBC_OFFLN KBC_ONLN	EQU EQU EQU EQU EQU EQU EQU EQU	10000010B 1000010B 10000101B 10000110B 10000111B 10001000	; Enable key click ; Disable key click ; Enter caps locked ; Exit caps locked ; Enable auto repeat mode ; Disable auto repeat mode ; Disable keyboard ; Enable keyboard ; Turn on offline LED
32 33 34 35 36 37 38 39 40 41	0082 0083 0084 0085 0086 0087 0088 0089 0089 008A 008B	KBC_EC KBC_DC KBC_EL KBC_ER KBC_DR KBC_DK KBC_EK KBC_OFFLN KBC_ONLN	EQU EQU EQU EQU EQU EQU EQU EQU	10000010B 1000011B 10000101B 10000110B 10000111B 10001000	; Enable key click ; Disable key click ; Enter caps locked ; Exit caps locked ; Enable auto repeat mode ; Disable auto repeat mode ; Disable keyboard ; Enable keyboard ; Turn on offline LED ; Turn off offline LED
32 33 34 35 36 37 38 39 40 41	0082 0083 0084 0085 0086 0087 0088 0089 0088 008B	KBC_EC KBC_DC KBC_EL KBC_ER KBC_DR KBC_DK KBC_EK KBC_OFFLN KBC_ONLN	EQU EQU EQU EQU EQU EQU EQU EQU	10000010B 1000011B 10000101B 10000110B 10000111B 10001000	; Enable key click ; Disable key click ; Enter caps locked ; Exit caps locked ; Enable auto repeat mode ; Disable auto repeat mode ; Disable keyboard ; Enable keyboard ; Turn on offline LED ; Turn off offline LED
32 33 34 35 36 37 38 39 40 41	0082 0083 0084 0085 0086 0087 0088 0089 008A 008B	KBC_EC KBC_DC KBC_EL KBC_XL KBC_ER KBC_DR KBC_DK KBC_EK KBC_DK KBC_EK	EQU EQU EQU EQU EQU EQU EQU EQU	10000010B 10000011B 10000101B 10000110B 10000111B 10001000	; Enable key click ; Disable key click ; Enter caps locked ; Exit caps locked ; Enable auto repeat mode ; Disable auto repeat mode ; Disable keyboard ; Enable keyboard ; Turn on offline LED ; Turn off offline LED
32 33 34 35 36 37 38 39 40 41	0082 0083 0084 0085 0086 0087 0088 0089 008A 008B	KBC_EC KBC_DC KBC_EL KBC_XL KBC_ER KBC_DR KBC_DK KBC_EK KBC_DK KBC_EK	EQU EQU EQU EQU EQU EQU EQU EQU	10000010B 10000011B 10000101B 10000110B 10000111B 10001000	; Enable key click ; Disable key click ; Enter caps locked ; Exit caps locked ; Enable auto repeat mode ; Disable auto repeat mode ; Disable keyboard ; Enable keyboard ; Turn on offline LED ; Turn off offline LED

			CDT ···	1	ouivalences
1			URI VI	ideo controler e	
3	0000	CRESET	EQU	00000000B	; Reset 8275 ; Normal rows, 80 chars/row
4	004F	SCRN1 SCRN2A	EQU	01001111B 10011000B	; Normal rows, 80 chars/row : 3 URTC, 25 chws/screen
5	0098 00DC	SCRN2B	······ĒQU	11011100B	; 3 VRTC, 25 rows/screen ; 4 VRTC, 29 rows/screen
7	0099	SCRN3	EQU	10011001B	; line 10 underline, 10 scan lines
······ ģ······	004E	SCRN4A	EQU	01001110B	; attributes, cursor, 30 HRTC
	004F	SCRN4B	EQU	01001111B	; attributes, cursor, 32 HRTC
10 11	0020	CSTRT	EQU	00100000B	; Start display, O delay, 1/burst
·····•	0040	CSTOP	EQU	01000000B	; Stop display
13	0080	CCRSR	EQU	10000000B	; Load cursor Position
14	0000	CINT	EQU	10100000B	; Enable end of frame interrupt
15	00E0	CSET	EQU	11100000B	; Preset counters
		• • • • • • • • • • • • • • • • • • • •			
		······································	· · · · · · · · · · · · · · · · · · ·		
······································	,				
		· · · · · · · · · · · · · · · · · · ·			
	· · · · · · · · · · · · · · · · · · · ·	••••			

)))))))

Z29ROM; MIC	ROBENCH 8051 CROSS AS	SEMBLER (V1)-218 24		5:30:06 PAGE 6	
ZH29.QQI	MPUTER.TERMINAL FIRMW	ARE.VERSION.1.03.			
<u>1</u>			Exter	nal memory address	map
	2000	DPATREG	EQU	8*1024	; 8275 data register
4 5	2001	DCMDREG	EQU	8*1024 + 1	; 8275 command register
	2001	DSTAREG	EQU	8*1024.+.1	; 8275 status resister
	3000	MLATCH	EQU	12*1024	; Memory latch
	4000	ATRIBMEM	EQU	16*1024	5 Attribute memory
11	5000	CHARMEM	EQU	20*1024	• Chanacter and
12 13	5780	L25MEM	ĒĞÜ	20*1024 + 1920	; Character memory ; Line 25 memory
14	57D0	IBUF	EQU	20*1024 + 2000	; Input buffer 32 bytes
15	0009	ILO	EQU	9	; Input queue lo water mark
16 17	0015	IHI	EQU	21	; Input queue hi water mark
18	0020	ILEN	EQU	32	; Input queue lenath
19	57F0	OBUF	EQU	20*1024 + 2032	; Output buffer 8 bytes
20 21	57F7	OLEN	EQU	0BUF + 7	; Output queue length
22 23	57F8	PMBUF	EQU	20*1024 + 2040	; ANSI parameter buffer 8 bytes
2 4 25	6000	DMAMEM	EQU	24*1024	; DMA memory
26 27	7000	EAROM	EQU	28*1024	5 EAROM memory
28	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		•••••	
29 30			Exteri	nal memory constan	ts
31	OOEF	ATRANL	EQU	11101111B	: Mask -CHARMEM- to -ATRIBMEM-
32	OOCF	DMAANL	EQU	11001111B	; Mask -DMAMEM- to -ATRIBMEM-
33	0010	CHRORL	EQU	00010000B	Mask -ATRIBMEM- to -CHARMEM-
34	0020	DMAORL	EQU	00100000B	; Mask -ATRIBMEM- to -DMAMEM-
			• • • • • • • • • • • • • • • • • • • •	•••••	
				•••••	
				• • • • • • • • • • • • • • • • • • • •	
				• • • • • • • • • • • • • • • • • • • •	
		•••••	*******		

	••••••				
					······
					••••••
					······
			• • • • • • • • • • • • • • • • • • • •	•••••	······································
		••••••	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	······································

1			Inter	nal RAM memor	y setup	
2		;				
4						
5			Resist	ter bank #0 u	sed in background jobs	
6 7	REG	; PTR1	EQU	RO	; Work pointer #1	(RO)
<u>ģ</u>	0000	PTRIA	ĒQŬ	Юон	; Address of -PTR1-	
9	REG	PTR2	EQU	R1	; Work pointer #2	(R1)
10	0001	PTR2A	EQU	01H	; Address of -PTR2-	
11	REG	INDX1	EQU	R2	; Index register 1	(R2)
…įį	0002	INDXIA	EQU	02H	; Address of -INDX1-	
13	REG	INDX2	EQU	R3	; Index resister 2	(R3)
1.4	0003	INDX2A	EQU	озн	; Address of -INDX2-	
	REG	WORK	EQU	R4	; Work resister	(R4)
···i6	0004	WORKA	EQU	04H	; Address of -WORK-	(05)
17 18	REG	TEMP	EQU	R5 	; Temporary holding register ; Address of -TEMP-	(R5)
	0005	TEMPA	EQU EQU	R6	; Address of TIEMPT ; Work register 1	(D/)
<u>19</u>	REG	WORK1 WORK1A	EQU	<u>ro</u>	; Address of -WORK1-	(R6)
20	0006 REG	WORKIA WORK2	EQU	R7	; Work resister 2	(R7)
22	0007	WORKZA	EQU	о ́л н	; Address of -WORK2-	
22	0007	WOINZH	200	0711) 11d at 6 2 2 1 440111142	
23 24				,		
25		;	Regis	ter bank #1 u	sed in DMA routine	
25 26		, , , , , , , , , , , , , , , , , , , 				
	DEC	DMPTR	EQU	RO	s Daimban imba lima andon	(RO)
2/	REG	DOT IN			; Pointer into line order	
27 28	KEG	DMPTRA	EQU	08H	; Address of -DMPTR-	
28	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	(R1)
2/ 28 29 30		DMPTRA	EQU	08H R1 09H	; Address of -DMPTR-	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	
28 29	0008 REG	DMPTRA DMLN	EQU EQU	08H	; Address of -DMPTR- ; Current DMA line number	

)

)

)

)

)

			Ģenera	al use variab	les from OAH to 1FH
2		; 			
3	000A	DMADRH	EQU	OAH	: Next address of DMA high byte
5		DMADRL	ENO	овн	; Next address of DMA low byte
6		ISTORE	EQU	OCH	; Input store pointer
	000D	IFETCH	EQU	ODH	; Input fetch pointer
8	000E	ICOUNT	EQU	OEH	; Input number count
				<u></u>	
10	000F	OSTORE	EQU	OFH	; Output store pointer
11	0010	OFETCH	EQU	1.0H	; Output fetch pointer
	0011	XPOS	EQU	11H	; Actual X-position
13 14	0012	YPOS	EQU	12H	; Actual Y-position
15					
16	0013	LINADH	EQU	13H	; Address of next character to plot
17 18	0014	LINADL	EQU	14H	; Address of next character to plot
18 19	0015	DSADRH	con	4 = 1.1	. Minnakah nadanan bink
······20	0016	DSADRL	EQU	15H 16H	; Dispatch address high ; Dispatch address low
21		20112112	200	1011	, pro-grand addition to
22	0017	TSCROLL	EQU	17H	; Top line of scrolling region
23 24	0018	BFIX	EQU	18H	; Top line of the bottom fixed region

25 26	0019 001A	PMADRH	EQU	19H	; ANSI parameter address high
27	001H 001B	PMADRL PMNUM	EQU EQU	1AH 1BH	; ANSI parameter address low ; ANSI number of parameters
·····	001c	PMVALUE			; ANSI current value of parameter
29 30		1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1			
	ÓOID	KBIDNUM	EQU	1DH	; Keyboard ID number
31 32	001E	HSLINE	EQU	1EH	; Hold screen mode line count
32	001F	ĞSET	EĠÜ	1FH	; Graphic set G1 and GO select
	• • • • • • • • • • • • • • • • • • • •				
	•••••				
••••••				• • • • • • • • • • • • • • • • • • • •	
•••••					
				•••••	
		······································			
		······································			
		······································			
••••					

729ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE 9 *** Z-29 COMPUTER TERMINAL FIRMWARE VERSION 1.03 *** Flags that define byte 20h ; External latch byte EQU 20H **XLATCH** 0020 ; O = save EAROM ; O = blink chars on **EAROMSAVE** BIT OOH 0000 1 = blanked BLINKF BIT бiн 00001 1 = auxiliary 1 = normal BIT 02H 0 = main portPORTE 0002 ; 0 = recall EAROMRCALL BIT OBH. 60003 Flass that define byte 21H 11 12 ; Attribute byte EQU 0021 ATTRIBUTES 13 14 08H ; Reverse video attribute RVVA BIT 15 0008 ; Underline attribute ΒΊŤ 09H 16 0009 UNDLNA ; Blink attribute BLINKA BIT OAH 17 000A ; Half intensity attribute HALFIA BIT ÖBH 18 OOOB ; Alternate character set attribute OCH **ALTCHARA** BIT 19 0000 1 = graphics ÖDH ; O = no graphics BIT GRPH 20 OOOD 1 = protected ; 0 = no protected PROTECT BIT 0EH 21 000E ; 0 = no alt graphics 1 = alt graphic set ΒÏΤ OFH 22 ÖÖÖF ALTGRPH 23 24 Flass that define byte 22H 25 26 ; O = no terminal XOFF XOFFSENT BIT 27 0010 1 = user XOFF 11H ; O = no user XOFF BIT 28 UXOFFSENT 0011 1 = G1 selected GSEL BIT 12H ; O = GO selected 29 0012 ; 0 = do not transmit 13H 1 = ok to trans ÖKTRANS BIT 30. 0013 ; O = no X-pos change 1 = changes 14H BIT DΧ 31 0014 ; 0 = line 25 disabled 1 = 25 enabled 15H BIT 32 0015 L25EN 16H ; O = no XOFF received 1 = received **XOFFRCVED** BIT 33 0016 34 35 Flags that define byte 23H 37 PSDF BIT ; Indicates if *PSD* has been done once 18H 38 0018 ; Indicates if *GNP* has been done once GNPF BIT 19H 0019 39 ; O = cursor disabled 1 = enabled BIT 1AH **ENBLCUR** 40 001A

1BH ERM BIT 41 001B ; shift data from keyboard BIT 1CH SHIFTF 001C control data from keyboard BIT 1DH CONTLE 43 001D ; O = no access from kb 1 = in progress 1EH BIT 001E ACCESSE ; O = keyboard enabled 1 = disabled KBDISF BIT 1FH 45 001F 47 Flass that define byte 24H ; Used in screen DMA must allays be O! 20H 0020 CLRFLG BIT 50 ; O = not insert char 1 = IC mode on BIT ICMODEF 51. 0021 ; O = dont hold screen 1 = stop screen BIT 22H 0022 **HSSTOPF** 52 ; O = not printing 1 = printing... BIT PRNTF .0023. 53 ; O = suppress video 1 = normal 24H VSPF BIT 54 0024 25H... BIT. .0025. GATM

	····		Flags	.that.define	byte 25H	
3	0028	; PRTYENA	D. I. T.	2011		
4	0029	PRTYSTK	BIT	28H 29H		1.=.ok.manity
5	092A	PRTYENN	BIT	2AH		1 = stick
6	002B	AUTOWRAP	BIT	297 2BH		
7	092¢	AUTOCR	BIT	20H		1 = wrap around
 	002D	AUTOLF	BIT	20n 2DH		r i = auto ur
9	QQ2E	ONLINE	BIT	2EH	0 = no auto LF on C	
10	002F	FULLDPLX		45. 7 2FH		
11	002F	FULLDELX	BIT	2FH	; O = half duplex	1 = full duplex
12		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
.13		,	Flags	.that.define	hyte 26H	
14		**************************************			. # 1.4 % * # 1.4	• • • • • • • • • • • • • • • • • • • •
.15	0030	KPADSHTF	BIT	30H	; O = keypad ordinary	1 = shifted
16	0031	KPADALTF	BIT	31H	; O = keypad ordinary	
.17	0032	CRUL	BIT	32H	5 0 = block cursor	
18	0033	CLKF	BIT	957 33H	; 0 = no key click	1 = click
.12	0034	REPF	BIT	34H		
20	0035	CAPLOCKF	₽.4 / BIT	9 4 0 35H		l = auto repeat 1 = locked
21	0036	HSMODEF	BIT	36H		
22	0037	L250N	BIT	37H	; 0 = not hold screen ; 0 = status line off	1 = status on
.23	000,	CZOON	DII	3/П	; U = Status line off	1 = status on
24		***************************************			•••••	• • • • • • • • • • • • • • • • • • • •
.25		:	Flage	.that.define	hv+a 27⊔	
26		***************************************			. W.V. & + V.O	• • • • • • • • • • • • • • • • • • • •
27	0038	CRNBK	BIT	38H		1
28	0039	FREQ	BIT	990 39H	; 0 = 50 Hz	1 = 60 Hz
29	003A	MONITOR	BIT	32H		
.77 30	003B	SCRNSAVE	BIT	ЭНЛ ЗВН		
.31	0030	PRTF2	BIT		; O = no screen saver	
32	903D			3CH		1 = auxilary
32 33	003E		BIT BIT	3DH	; O = normal char set	
34 34	······›	HNDSHK		ЗЕН		1 = hardware
35						
36	• • • • • • • • • • • • • • • • • • • •	***************************************		that define	L. L. 000	
37		,	LIGS	that derine	DYTE ZOM	
7/ 38	0028	MODE	EQU	28H	; Mode terminal is in	
39	0020	HODE	EGO	2011	; mode terminal is in	
40	0040	ANSI	BIT	40H		1 _ ONCT 3 -
41	0041	ZDS	BIT	41H.	O = other modes	1 = ANSI mode
 42	0042	ADM3	BIT	42H	; O = other modes	1 = ZDS mode
43	0043	HAZ	BIT		; O = other modes	1 = ADM 3 mode
79 44				43H		1 = Haz 1500
45						
46			Vaniak	ole tab table	· · · · · · · · · · · · · · · · · · ·	•••••
47 48		•	vai 141			
48	0029	TABTAB	EQU	29H	; Variable tab table	10 bytes to 32H
					, , , , , , , , , , , , , , , , , , , ,	TO DITES TO OZII
			• • • • • • • • • • • • • • • • • • • •		•••••	•••••
• • • • • • • • • • • • • • • • • • • •		•••••				
	************************		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	•••••	

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE 11 *** Z-29 COMPUTER TERMINAL FIRMWARE VERSION 1.03 *** from 33H to 46H ; Serial port baud rate
; Non-volatile RAM checksum EQU BAUDRATE 0033 NVRSUM ÉQÙ. 0034 ; Saved X-position XXPÓS ÉQU 35H 0035 ; Saved Y-position 36H XYPOS EQU 0036 ; Blinking rate of screen ; Running count for blink rate EQU 37H 0037 BLNKRATE EQU 38H 10 0038 BLNKCNT ĖQÚ. 39H ; Auto off timer count 12 0039 AUTOCNT 13 ĖQU ЗАН 14 OOBA TONT ; Running clock count 15 TSEC EQU звн ; Seconds counter 003B 3CH ; Minute counter 16 ,003¢, TMIN ÉQÚ EQU 3DH ; Hour counter 17 003D THOUR 18 EQU ЗЕН ; Last count of seconds displayed 19 DISPSEC 003E 20 EQU 3FH ; Protected fields byte PFIELD 21 003F 22 0040 ERRORS ĖQÚ. 40H ; Powerup diagnostic error flags Screen line table 27 28 LORDER 30 Stack area 31 005F STACK ĖQÜ 5FH ; Stack pointer address 32 bytes to 7FH 32 33 35 8051 processor bit port defines 36 KBOUT BIT Keyboard output line 37 0090 BIT P1.1 ; Keyboard input line KBIN 0091 38 O = normal video P1.2 VSP BIT 39... 0092 P1.3 ; O = setup mode locked 1 = SETLCK BIT 0093 P1.4 ; Data terminal ready 0094 DTR BIT P1.5 ; Ready to send RTS BIT 42 0095 BIT P1.6 ; Expansion socket 43 CLKRUN P1.7 ; Clock run for CRT controler BIT 44 0097 45 P3.1 ; Predefined serial port transmit pin TXD BIT 46 DMATYPE P3.4 ; O = clear memory 47 00B4 BIT P3.5 ; Clear to send 00B5 CTS BIT

<u>1</u>		ROM. #2 .VECTORS
2 34	 ;	These locations are provided to facilitate future modifications without a change to the masked processor. Each location is
.56		accessed.with.a.=CALL=.and.control.is.gained.back.with.a.=RET=.
7 8 9 	5 	Initialization of internal RAM and any other devices that may be added,
1000	; R2_IN	EQU 1000H
.2 .3		
.4 .5 .6	ş 	Same as R2_IN but does not destroy current settings such as CAPS LOCK, ON/OFF LINE, TIME on clock, etc.
7 1003 8	R2_IN2	EQU 1003H
9	•••••	
20 21 22		Start DMA hook. This is jumped to at the beginning of each DMA request from the CRTC.
23 1006 24	R2_STARTDMA	EQU 1006H
25 26		····· <u>·</u> ·······
:6 ?7 !8		Stop DMA hook. This is jumped to at the beginning of each timer O interrupt.
9 1009 0	R2_STOPDMA	EQU 1009H
1	•••••	
2 3 4		Fill screen hook. This is jumped to start the fill screen DMA feature. A check for protected fields is made.
5 100C	R2_FILL	EQU 100CH
 8		
9	; ;	IRQ hook. This is jumped to at the beginning of each IRQ.
0 100F 1	R2_IRQ	EQU 100FH
2 3 4	,	Serial hook. This is jumped to at the beginning of each serial interrupt.
5 6 1012	; R2_SERIAL	EQU 1012H
••••••••••••		
•		
	•••••	

<u>1</u>			Transmit hook. This performs the transmit character function of the terminal for the serial port.
3 4 5	1015	; R2_XMT	EQU 1015H
			Send an XON to the host.
8 9 	1018	R2_XON	EQU 1018H
11			Initialize the cursor format.
13 14 15	101B	; R2_ICUR	EQU 101BH
16 17			Initialize the keyboard
18 19 20	101E	, R2_IKB	EQU 101EH
21 22			SETUP mode vector
23 24 25	1021	R2_SETUP	EQU 1021H
26 27 28			Background loop hook. This is called once every time through the main background loop.
29 30	1024	; R2_BACK	EQU 1024H
31 32 33			Keyboard input hook. This is called once before a character
3 4 35 36	1027	; ; R2_KY1	is received from the keyboard. EQU 1027H
37 38 39			Keyboard output hook. This is called once at the beginning
40 41		; ;	of each transfer to the keyboard.
42 43	102A	R2_KY2	EQU 102AH
44 45 46			Escape parsing hook. This is called after an -ESC- is received.
47 48	102D		EQU 102DH
	······································		

1		;	Transmit character. This performs the function of
2 3		\$ •	transmitting the character at the current position.
4 5	1030	R2_TC	EQU 1030H
6		·····	
/ 8	• • • • • • • • • • • • • • • • • • • •	······································	Transmit line. This performs the function of transmitting the line at the current position.
9 10 11	1033	, R2_TL	EQU 1033H
12	• • • • • • • • • • • • • • • • • • • •		
.13 14			Transmit page. This performs the function of transmitting the page to the port that is not selected.
15 16 17	1036		EQU 1036H
18 .12		······································	Transmit 25th line. This performs the function of transmitting
20 21			the 25th line to the port that is not selected.
 22 23	1039		EQU 1039H
24 25		\$	Print page. This performs the function of transmitting the
26 .27		; ;	<pre>page to the opposite port in a form proper for printing (uses CR, LF at the end of each line).</pre>
28 29 30	103C	; R2_PRNT	EQU 103CH
31			
32 33		; ;	Display clock. This updates the time on the 25th line.
34 .35	103F	R2_DCLK	EQU 103FH
36 .37 .38			Display 25th status line. This displays the system status
38 39 40		; 	on the 25th line if the option is enabled and the user does not have the 25th line enabled.
40 .41	1042	; R2_DISP	EQU 1042H
		•••••	
	•••••		

5			·····	····· <u></u> ···			
6 			; ;		vector		
8 9 0000 10	80	2E		ORG SJMP	0000Н START	; Start of program	
				Éxtern	nal interrupt reques	(
13 14	9993			ÖRĞ	0003Н		,
15 0003	30	AF 02 22		JNB	EA, I1	; Double check	
16 0006 17 0008 18 000A	80 D2 32	22 89	I1:	SJMP SETB RETI	IEXTIO TCON.1	; Start DMA transfer	
<u>19</u>				Timon	O interrupt vector		
21 22						•••••	
23 24 000B	000B	AF 02		ORG JNB	000ВН ЕА, I2	* Model of Charles	
25 000E	80			SJMP	ITMRO	 Double check stop DMA transfer 	
25 000E 26 0010	D2	<u>1</u> D	12:	SETE	TCON.5		• • • • • • • • • • • • • • • • • • • •
27.0012 28	32			RETI			
. 29 30 31			·····	Extern	al interrupt reques	t i	
3 1	0013	• • • • • • • • • • • • • • • • • • • •		ORG	0013H	•••••••••••••••••••••••••••••••••••••••	
33 0013	30 02	AF 03		JNB	EA, I3 R2_IRQ	; Double check	
34 0016	02	100F	***	LJMP	R2_IRQ	; Interrupt request	
35 0019 36 001B	D2 32	SB.	I3:	SETB RETI	TCON.3		
.37 .38		•••••					
.39 40				Serial	port interrupt vec	tor	
. 41	0023 30	AF 03	·····		0023H EA, I5		
42 0023 43 0026	30 02	AF 03		JNB LJMP	EA, I5 R2_SERIAL	; Double check	
44 0029 45	32		I5:	RETI		; Serial Port	
46 47			• · · · · · · · · · · · · · · · · · · ·	limpe	to.other.locations		
48			**************************************			••••••	
.49.002A	02 02	1004 1009	IEXTIO: ITMRO:	LJMP LJMP	R2_STARTDMA R2_STOPDMA	; Start DMA jume ; Stoe DMA jume	

<u>.</u>					Initia	lize.system	
				; 	Set.st	ack.mointer.and.call.	routine to set up everything
i	0030				ORG	зон	; Somewhat arbitrary
. 0033 . 0033	75 12	1000	5F	START:	LCALL	SP, #STACK R2_IN	; Initialize stack pointer ; Initialize everything else
) !					MAIN -	main control loop	
				; ;	*MAIN* whenev	is in charge of givier there is a charact	na control to the proper routine er present in one of the FIFOs.
0036	12	1024		MAIN:	LCALL	R2_BACK	; Background hook
0039 003C 003E	20 31 40	22 F6 02	06		JB ACALL JC	HSSTOPF, MN1 FCIF MN1 IFCP	; IF not hold screen THEN ; fetch char from input FIFO ; IF character THEN ; processes character
0040 0042 0045	30 30	79 2E 13	20 1D	MN1:	ACALL JNB JNB	ONLINE, MN2 OKTRANS, MN2	; processes character ; IF online AND ; IF output port free THEN
0048 004B 004E	20 30 20	02 3E B5	03 03 14	MN1A:	JB JNB JB	PORTF, MN1A HNDSHK, MN1B CTS, MN2	; IF auxiliary port OR ; IF hardware handshake THEN ; IF not CTS THEN skip
0051 0054 0056	20 51 40	16 4C OD	11	MN1B:	JB ACALL JC	XOFFRCYED, MN2 FCOF MN2	; IF not -XOFF- received THEN. ; fetch char from output FIFO ; IF character THEN.
0058 005B 005D 0060	B4 D2 B4	13 11 	02 02	MN1C:	CJNE SETB CJNE CLR	A, #XOFF, MN1C UXOFFSENT A, #XON, MN1D UXOFFSENT	; IF char = xoff THEN ; flag user xoff sent ; IF char = xon THEN ; clear user xoff sent
0062	C2 12	1015		MN1D:	LCALL	R2_XMT	transmit character
0065 0068 006A	20 51 40	1F B6 02	06	MN2:	JB ACALL JC	KBDISF, MN3 FCKB MN3	; IF keyboard enabled THEN ; fetch char from keyboard ; IF keyboard data THEN
.006C	51	EE			ACALL	KBCP	; process keyboard data
006E 0071 0074	30 20 12	37 15 103F	C5 C2	MN3:	JNB JB LCALL	L250N, MAIN L25EN, MAIN R2_DCLK	; IF 25th status enabled AND ; IF 25th line disabled THEN ; display clock every second
0077	80	BD	•••••		ŞJMP	MAIN	; Go forever

1					IFCP -	input FIFO character	Processor
2 3				7	* TECD*	intonopote character	
 4	• • • • • • • • • • • • • • • • • • • •				struck	on the keyboard if o	s which have been received (or ff-line or keyed as a local
<u>5</u>					fyncti	on). If the characte	r is part of an escape sequence
6 7				;	then c	ontrol is given direc	tly to that routine by means of it is a control character then it
ģ	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	is loo	ked up in *CTLTAB*.	If found the program jumps to the
9					approp	riate routine, otherw	ise the character is discarded. If
0 1				;	it is	a displayable characto	er it is placed on the screen mode, attributes, etc.
ž Ž					arter.	checking for anaphics	mode, attributes, etc.
3 4				· · · · · ,	ENTRY	(A) = character from	n input FIFO
5 6					ĖXIT	none	
7. 					ÜSĖS		***************************************
8 9				,	USES	all	
0 1 0079	B4	18	06	IFCP:	CJNE	A, #CAN, IFFA	; IF canceled AND
2 007C	20	<u>1</u> 8	·····ŏš····		JB	HAZ, IFPA	; IF not Haz mode THEN
3 007F	12	082F			LCALL.	SETNORM	; Restore dispatch address
7 5 0082	CO	16		IFPA:	PUSH	DSADRL	; Jump to the
6 0084	çö	15			PUSH	DSADRH	; -dispatch address-
7_0086 8	22			•••••	RET		••••••
9.0087	B4	OD.	12	NORM:	CUNE	A, #CR, IFP1	; IF carriage return THEN ; IF hold screen mode THEN
0 008A 1 008D	30 E5	36 12	OF		JNB	HSMODEF, IFP1 A, YPOS	; IF hold screen mode THEN
2 008F	04	· · · · · · · · · · · · · · · · · · ·	•••••	• • • • • • • • • • • • • • • • • • • •	INC	A 1509	; get y-pos to check
3 0090	B5	18 	00		CJNE	A, BFIX, NRMO	; with bottom fixed region
4 0093 5 0095	40 D5	05 1E	02	NRMO:	JC DJNZ	NRM1 HSLINE, NRM1	; IF YPOS >= bottom THEN
6 0098 8 0098	<u>D</u> 2	<u>‡5</u>	02		SETB	HSSTOPF	; decrement line ; IF zero THEN stop disp
7 009A	74	OD		NRM1:	MOV	A, #CR	restore carriage return
8 9				;	Check	for monitor mode and c	ontrol characters
0		~~	on.	; 			
1 009C. 2 009F	30 	3A	op	IFP1:	JNB	MONITOR, IFP1B	; IF monitor THEN ; start with normal display
3 00A1	B4	20	00		CUNE	A, # , IFP1A	, start with normal display
4 00A4	50	5E		IFP1A:	JNC	IFP6	; IF control character THEN
5.00A6. 6.00A8	D2	98			SETB	RVVA	
00A8 7 00AA	80	58			ORL SUMP	A, #01000000B IFP6	<pre>; convert to ordinary char ; skip ahead</pre>
8							
9 00AC	<u>B4</u>	20 03	00	IFP1B: IFP1C:	CUNE	A, #1 (, IFP10 IFP2	
1 00B1	94 B4	7F	21	1FF1C+	CUNE	A, #RUBOUT, IFP3	; IF control character OR
2 3						ter is a control chara	
4 5 00B4	• 75	EO	00	;			
9.0084 6 0087	75 90	F.Q 0119	Y.9	IFP2;	MUY MOV	B, #CTAB1L DPTR, #CTAB1	
	20	40	12		JB		

🚣 . 🦫	OOBD	20 75	41	OF		JB	ZDS, IFP2A	; IF not ZDS mode THEN ; get other table info
	0003	75 90	F0 0107	0E		MOV MQV	B, #CTAB2L DPTR, #CTAB2	; get other table info
	9006 9006	79 20	9197 42	 06			ADM3, IFP2A	; IF not ADM 3 mode THEN
	00C9	75	F.Q	9.6		YQV		; get other table info
7 (0000 000F	90 31	0128 3A		IFP2A:	MOV ACALL	DPTR, #CTAB3 STAB	; Search the table
8								
10			• • • • • • • • • • • • • • • • • • • •			Çontro Subrou	l.character.found.in tine return will caus	table, go to associated address. se return to main idle loop
12 (OOD1	40	33		?	JC	IFPRET	; IF found THEN
	QQD3 OOD4	E4 73				ÇLR JMP	A @A+DPTR	; jump to address in (DPTR)
15								
16 17					;	Charac	ter is a displayable	character
18 (00D5	30	43	06	; IFP3:	JNB	HAZ, IFP3A	; IF Hazeltine mode AND
12.9		B4	7E	Q3		ÇUNE	A, #1~1, IFP3A	; IF char = /~/ THEN
	OODB QODE	02 30	067B	13	TEDOA	LJMP JNB	ESCCODE GRPH, IFP5	; do escape stuff ; IF in graphics mode THEN
	ООЕ1	B4	9£ 5E	06	ACE.9H*	CJNE	A, #5EH, IFP4	
	QQE4	20	QF	Q.D		JB	ALTGRPH, IFP5	; IF alternate THEN dont map
24 (00E7	74	7F			MOV	A, #7FH	; IF 5EH THEN map to 7FH
	9.0E9	D3				SETB	Ç	
	OOEA	40 54	08 1F		IFF4:	JC	IFP5	; IF in graphics range THEN
	QQEC OOEE	94	አተ OF	03		ANĻ JNB	A, #00011111B ALTGRPH, IFP5	; mask to graphic character;; IF alt graphic set THEN
	00EE	D3		03		SETB	C	set carry and
	00F2	80	02			SJMP	IFP5A	;
32 33		· · · · · · · · · · · · · · · ·			:	Check	for insert character	made
34			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • •	***************************************			
	00F4	A2	QÇ E7		IFP5:	MOV	C. ALTCHARA	; Get alt char mode
36 37	00F6	92	E7		IFP5A:	MOV	ACC.7, C	; Place into data byte
9/	00F8	30	21	09	• • • • • • • • • • • • • • • • • • • •	JNB	ICMODEF, IFP6	; IF insert character mode THEN
	OOFB	co	EO			PUSH	ACC	; save character
	OOFD	31	A9			ACALL	SPF	; skip protected fields
	00FF 0102	12	ODC1			ÇALL POP	PIC ACC	; insert blank character ; restore character
43	0102	ьо				FOF	HCC	, restore character
44 (0104	31	57		IFP6:	ACALL	PUTCHR	; Put character onto screen
459	0106	22	• • • • • • • • • • • • • • • • • • • •		IFPRET:	RET		
	• • • • • • • • •							
			• • • • • • • • • • • • • • • • • • • •					

C

	1			;;	CTAB1/2	/3 — control character ta	bles
	2 3			; ;	*CTAB1*	contains all control cod	les for the ANSI and ZDS
	4 			7	emulati	on modes.	
•••••	<u>5</u>	• • • • • • • • • • • • • • • • • • • •				contains all control cod	es for the ADM 3 emulation
•••••	7 .			· • · · · · · · · · · · · · · · · · · ·	mode.		
	9				*CTAB3*	contains all control cod on mode.	es for the Hazeltine 1500
	11			. į			
	13			; ;	Table en	ntries are: The control of the routine which per	character followed by the forms the requested function.
••••••	14		·····	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		
••••••	16	• • • • • • • • • • • • • • • • • • • •	0107	CTAB2	ÉQÚ	\$	••••••
	17 18	0107	ÓB		DB	овн	; Vertical tab
• • • • • • • • • • • • • • • • • • • •	19 20	0108	OC8D		₽₩	CUP	; Cursor up
	21	010A	ос		DB DW		; Form feed
	22 23	0108	0C64			CRT	; Cursor right
	24	OTOD	OE .	• • • • • • • • • • • • • • • • • • • •	DB		; Shift out
	25	010E	OFB3		DW	EKI	; Enable keyboard input
	27.	0110 0111	OF B9		DB DW	OFH BKI	; Shift in ; Disable keyboard input
	29						
		0113 0114	1A OBAC		DB DW		; Substitute ; Clear screen
	32			• • • • • • • • • • • • • • • • • • • •			
		0116 0117	1E 0C36		. DB 		; Record separator ; Set cursor home
	.35 .36		0119	CTAB1	EQU	\$	
	37						
		0119 011A	1B 067B		DB DB		; Escape ; Escape code processor
	40						
		011C 011D	0C3D		. DB 		; Backspace ; Perform backspace
	43	011F			DB	нт	; Horizontal tabulation
	45		05C1	• • • • • • • • • • • • • • • • • • • •	DW		; Tab to next column
	46	0122	0E		DB	0EH	; Shift out
	48	0123	OF81	• • • • • • • • • • • • • • • • • • • •	DB DW		; Pick G1 character set
	49. 50	0125	ÖF		DB	OFH	; Shift in
			OF7B				Pick GO character set
	53		0128	CTAB3	EQU	.\$	
	54 55	0128	OA.		DB	LF	; Line feed
			0555		DB DW	PLFCR	; Perform LF and/or CR

			MARE.YERSION.1.03.**	pB	ÇR	
4	l 0128 2 0120 3			DW	PCRLF	; Perform CR and/or LF
4	1 012E	07		DB	BELL	; Bell
	591,2F 5	Q66B		DW	RING	; Ring the bell
	7	0008 000E	CTAB1L CTAB2L	EQU EQU	(\$-CTAB1)/3 (\$-CTAB2)/3	; Table lenath ; Table lenath
	7		UTABZL	E80		
10	0131 0132	08 0C51		DB DW	BS HCLFT	; Back space ; Cursor left
12	· · · · · · · · · · · · · · · · · · ·		•••••			
13	0134 0135	10 0079		DB	10H HCRT	; Data link escape ; Cursor right
15	5 0137	09			HT	; Horizontal tabulation
17	7 0138	0636		DB DW	HTAB	; Morizontal tabulation ; Special function
18 19	3	0006	CTAB3L			; Table length
	·		CIMPOL			
	• • • • • • • • • • • • • • • • • • • •					

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;		he first character i (A) = character to (B) = table length (DPTR) = table add (A) = original cha (B) = number of en (DPTR) = first add	match in number of entries ress racter tries left to search + 1 ress after character match tch, clear otherwise ; Save character to match
ST1:	ENTRY EXIT USES MOV CLR	(A) = character to (B) = table length (DPTR) = table add (A) = original cha (B) = number of en (DPTR) = first add 'C' = set if no ma A, B, DPTR, WORK WORK, A	match in number of entries ress racter tries left to search + 1 ress after character match tch, clear otherwise ; Save character to match
ST1:	EXIT USĖS MOV CLR	(B) = table length (DPTR) = table add (A) = original cha (B) = number of en (DPTR) = first add C' = set if no ma A, B, DPTR, WORK WORK, A	in number of entries ress racter tries left to search + 1 ress after character match tch, clear otherwise ; Save character to match
ST1:	EXIT USĖS MOV CLR	(DPTR) = table add (A) = original cha (B) = number of en (DPTR) = first add (C' = set if no ma A, B, DPTR, WORK WORK, A	ress racter tries left to search + 1 ress after character match tch, clear otherwise ; Save character to match
ST1:	USÉS MOV CLR	(A) = original cha (B) = number of en (DPTR) = first add (C' = set if no ma A, B, DPTR, WORK WORK, A	racter tries left to search + 1 tries after character match tch, clear otherwise ; Save character to match
ST1:	USÉS MOV CLR	(B) = number of en (DPTR) = first add 'C' = set if no ma A, B, DPTR, WORK	tries left to search + 1 ress after character match tch, clear otherwise ; Saye character to match
ST1:	MOV CLR	C' = set if no ma A, B, DPTR, WORK WORK, A	tch, clear otherwise ; Save character to match
ST1:	MOV CLR	WORK, A	
ST1:	CLR		
ST1:	CLR		
		Α	
	MOVC	A AA, DDTD	; REPEAT ; get character from table
**************************************	INC	DPTR	; bump to address high byte
4 05	CUNE	Á, WORKA, ST2	; compare char with work
	CLR	A A, @A+DPTR	; IF character = work THEN
	MOVC	A, @A+DPTR	;
o	PUSH	ACC	; save on stack ; bump to address low byte
			; Dump to address low byte
		A. @A+DPTR	; get address low byte
2	MOV		; save address low byte
3	POP	DPH	; pop address high byte
	MOV	A, WORK	; restore orisinal character
		С	; clear 'C' for ok
	REI		; return
ST2:	TNC:	DPTR	; bump to address low byte
······································	INC	DPTR	; bump to next entry if any
O E7	DUNZ	B, ST1	; UNTIL #entries = 0
	MOV	A, WORK	; Restore original character ; Flas 'C' for no match
		C	; Flag 'C' for no match
	2 3 ST2: 0 E7	3 POP MOV CLR RET ST2: INC INC	CLR A

.1	· · · · · · · · · · · · · · · · · · ·				PUTCHR	put .chanacteronto.	screen
2				;	*PHTCH	R* puts the character	onto the screen and checks for
				*	wrap a	round at the end of a	line for auto CR or LF.
6 7				\$ \$	ENTRY	(A) = character	
. 9				; ,	EXIT	none	
10 11				;	USES	A, B, DPTR, TEMP, PI	TR1
12 13 14 0157							
14 0157 15	D2	14		PUTCHR:	SETB	DΧ	; Flas that cursor moved
16 0159 17	co	EO			PUSH	ACC	; Save character
18 015B 19	Зi	A9			ACALL	SPF	; Skip any protected fields
20 015D 21 0160	85 85	14	82 83		MOV MOV	DPL, LINADL DPH, LINADH	; Get line address
22 0163 23 0166	53 E5	<u>13</u>	83 EF		ANL MOV	DPH, #ATRANL A, ATTRIBUTES	; Mask to attribute memory
24 0168	FÖ	21			MÓVX	@DPTR, A	; Output current attributes
25 0169 26 0160	<u>43</u>	83 È0	10		ORL POP	DPH, #CHRORL ACC	; Mask to character memory ; Restore character
27 28 016E	F0				MÖVX	@DPTR, A	; Store into screen memory
29 30 016F	E5	ii			MOV	A, XPOS	
31 0171 32 0174	B4	4F 2B	09 		CUNE	A, #MAXCHAR, PU1	; IF cursor at end of line AND ; IF auto wrap on THEN
33 0177 34 0179	B1	5C 61			ACALL ACALL	PCR PLF	; perform carriage return ; perform line feed
35 017B	80	09			SJMP	PU2	; exit out
36 37 O17D	05	11		PU1:	INC	XPOS	; Bump -XPOS-
38 017F	A3				INC	DPTR	; Bump location in memory
39 0180 40 0183	85	82 83	14 		MOV	LINADL, DPL LINADH, DPH	; Save new line address
71							
42 0186	2i	A9		PU2:	AJMP	SPF	; Skip any protected fields

(

1						CPF -	check protected fields	5
2 3					;	*CPF*	thecks to see if the o	current position is protected.
<u>4</u>					*	If so	it returns a value in	the carry flas.
6					; ;	FNTER	XPOS and YPOS at Pos	sition
					;		(C) = set if protect	
10		••••••	• • • • • • • • • • • • • • • • • • • •		*			
12			• • • • • • • • • • • • • • • • • • • •		?		. A, B, DPTR, TEMP, WO	2003. F. (1) 1
13 14 0	188	E5	ЗF	• • • • • • • • • • • • • • • • • • • •	CPF:	MOV	A, PFIELD	; Get protected fields
15.0	18A	70 C3	02		CPFR1:	JNZ	<u>C</u> PF1	; IF none protected THEN
16 0					CPFR1:	CLR	C	; clear carry
17.0 18	iish	22				RET	· · · · <i>,</i> · · · · · · · · · · · · · · · · · · ·	i and return
12.0	18F	FC			CPF1:	MOV	WORK, A	; Save value
20 0	18F	F.C E5	11			MOV	A, XPOS	
210	121	85	12	FQ		MQV	B, YPQ\$	
22 0 23	194	B1	03			ACALL	CLA	; Calculate line address
24 0		53	83	EF		ANL	DPH, #ATRANL	; Mask to attribute memory
. 25.0	199	.E0				MQYX	A, @DPTR	
26 0 27	19A	54	OF			ANL	A, #00001111B	; Mask to get only attributes
4/ 28 0	19C	70	05			JNZ	CPF2	; IF no attributes set THEN
299		BC	80	EB		CUNE	WORK, #80H, CPFR1	; IF not protected THEN return
30 0	1A1	80	04			SJMP	CPFR2	; ELSE return protected
310	1A3	5 <u>.</u> ç			CPF2:	ANL	A,WQRK	; ELSE mask with protected
32 0		B5	04	E5		CUNE	A, WORKA, CPFR1	; IF no match THEN return
33 34 0		D3	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	CPFR2:	SETB	с	; Set carry
35.0	1A8	22.				RET		; Return
			•••••					
								••••••

))))))))))))))

	rear and the contract of the c	. /	RMWARE VERSION 1.03 **			
<u>1</u>		• • • • • • • • • • • • • • • • • • • •		SPF	skip over protected fie	lds
		• • • • • • • • • • • • • • • • • • • •	<u>\$</u>			rrent position is protected.
5			; 			d to the next unprotected sen. If there are no unprotected
6 7			,	Positio	ons from the current cu	rsor position to the end of the ed to the HOME position.
ś			······*	The scr	reen address is only up	dated if the cursor Position
9 10			,	changes	5.•	
			<u></u>	ENTRY	 none	
13			*			
14 15			; ;	EXIT	none	
15 16	•••••		**************************************	ÜŚĖŚ	A, B, DPTR, WORK, TEM	P, PTR1
17 18	• • • • • • • • • • • • • • • • • • • •					
19 01 20 01	LA9 31 LAB 40	88 01	SPF:	ACALL	CPF SPO	; Check protected fields ; IF not protected THEN
21 01			***************************************	RET		; return
22 23 01	IAE ES	11	SPO:	MOV	A, XPOS	
24 01 25 01	IBO B4		11	CJNE MOV	A, #MAXCHAR, SP2	; IF end of line THEN
26 01	1B6 E5	12	00	MOV	XPOS, #0 A, YPOS	; beginning of line
27 01 28 01	LBS B4 LBB 75		05	CJNE	A, #MAXLINE-1, SP1 YPOS, #0	; IF end of screen THEN ; beginning of screen
29 01				AJMP	BLINAD	; new address and return
30 31 01	ICO 05	<u>12</u>	SP1:	INC	YPOS	; ELSE bump line number
32 01 33	.c2 80	02		SJMP	SP3	; calc address and so
34 01			SP2:	INC	XPOS	; ELSE bump column number
35 01 36 01		EA DF	SP3:	ACALL SUMP	BLINAD SPF	; calc new address ; keep on going
					• • • • • • • • • • • • • • • • • • • •	
			•••••••••••		• • • • • • • • • • • • • • • • • • • •	
						.,
		***************************************	***************************************			
• • • • • • • • • • • • • • • • • • • •	· · · · · · · · · · · · · · · · · · ·	• • • • • • • • • • • • • • • • • • • •				
• • • • • • • • • • • • • • • • • • • •						
					, 4	
		•••••				

1					PCIF -	Place character in i	neut FIFO
2 3				; ;	*PCIF*	places a single char	acter into the input FIFO.
4 5				=			
6					ENTRY	(A) = character	
7 8	• • • • • • • • • • • • • • • • • • • •				EXIT	(A) = character	
9 10						'C' = set if no roo ''FO' = set if XOFF	m, clear otherwise is needed, clear otherwise
11 12 13					······································	A B, DPTR	
14		۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰		 pric•	CLR	EA	; Lock out entries
15 01CA 16 01CC	C2 75	AF 83	57	PCIF:	MÓV	DPH, #HIGH (IBOF)	
17 01CF 18 01D2	85 F0	oc	82		MOV MÓVX	DPL, ISTORE @DPTR, A	; Load input buffer pointer ; Write character into FIFO
19 01D3	. FŠ	F0			MOV	в, а	; Save temporary
20 21 01D5	E5	OE			MOV	A, ICOUNT	
22 01D7 23 01DA	B4 D3	20	03		CJNE SETB	A, #ILEN, PI1 C	; IF queue is full THEN ; set 'C' for no room
24 01DB	80	14			SJMP	PIR	; set 'C' for no room ; and exit out
25 26 OIDD		D5		PI1:	CLR	FO	; No -XOFF- until otherwise
27 01DF 28 01E2	B4 D2	15 D5	92		CJNE SÉTB	A, #IHI, PI2 FO	; IF queue at XOFF point THEN ; flas that -XOFF- is needed
29 30 01E4	5	OE		PI2:	INC	ICOUNT	; Bump count
31 01E6	05	oc 			INC MOV	ISTORE A, ISTORE	; Bump store and check wrap
32 01E8 33.01EA	E5 B4	F0 OC	03 		CUNE	A, #LOW (IBUF+ILEN)	, PI3
34 01ED 35	75	oc	DO		MOV	ISTORE, #LOW (IBUF)	
36 O1FO	C3	۸۳		PI3:	CLR SETB	C	; Flas everythins OK ; Enable entries
37 01F1 38 01F3	D2 E5	AF F0		PIR:	MOV	EA A, B	; Restore character
39.01F5	22				RET		

					FÇIF	.fetchchanastenfro	m.ineut.FIFO
2 3				,	*FCIE*.	.is.used.to.fetch.a.	single_character_from_the_ingut
4 =				7		f any are available.	
6				*			
7 8					EŅŢŖY	none	
9 10					ÉXIT	(A) = character if	available aracter, clear otherwise
. 1.1							aracter, crear otherwise
12 13				,	USES	A, B, DPTR	
14					MOUL	A TOOLINT	
15 01F6 16 01F8	E5 70	OE		FCIF:	MOV JNZ	A, ICOUNT FI1	; IF no data in queue THEN
17 01FA	D3 22				SETB		; set carry and return
18 01FB 19				•	RET		
20 01FC	B4	09	03	FI1:	CJNE	A, #ILO, FI2	; IF low water mark THEN
21 01FF 22	12	1018.			LÇALL	R2_XON	; send -XON- now
.23 0202 .24 0204	C2 75	AF 83	57	F.I.2	CLR	EA	; Lock out entries
24 0204 25 0207	75 85	83 OD	57 82		MOV MOV	DPH, #HIGH (IBUF) DPL, IFETCH	; Load input buffer pointer
26 020A	85 E0		×+		MOVX	A, @DPTR	; Read character from FIFO
.27 .020B 28	F.5,	F.Q			₩0Y	B , A	; Save temporary
29 020D	05 E5	OD OD			INC	IFETCH	; Bump fetch and check wrap
30 020F		OD	20		MOV	A, IFETCH	
31 0211 32 0214	B4	FO	03	• • • • • • • • • • • • • • • • • • • •	CJNE	A, #LOW (IBUF+ILEN IFETCH, #LOW (IBUF) ; Reset value of fetch pointer
33 34 0217	15	ÖE		F13:			; Decrement count
35 0217	15 21	FO		F13:	DEC AJMP	ICOUNT PI3	; Decrement count ; Return (saves ROM)
	,		• • • • • • • • • • • • •				
• • • • • • • • • • • • • • • • • • • •		•••••					
		• • • • • • • • • • • • • • • • • • • •					
		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			,	
• • • • • • • • • • • • • • • • • • • •							
. 							
	•						
• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •				
		• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •			

<u>1</u>					LCIF -	look at next characte	er on input FIFO	
2 3				;			racter from the input FIFO if there	
4 5					is one	but does not take it ahead´ applications.	off the FIFO. This is needed for	
6 7				; ;				
9				•	ENTRY	none		
9 10 11 12				**************************************	EXIT	(A) = character if a 'C' = set if no char	available racter, clear otherwise	
12 13				; ;	USES	A, DPTR		
13 14						::: T :::::	•••••••••••••••••••••••••••••••••••••••	
15 16 021B	E5	öĖ		LCIF:	MOV	A, ICOUNT		
17 021D 18 021F	70 D3	02			JNZ SETB	LI1	; IF no data in queue THEN ; set carry and return	
19 0220	22				RET			
20 21 0221 22 0224	75 85	83 	57	LI1:	MOV	DPH, #HIGH (IBUF) DPL, IFETCH		
22 0224 23 0227		OD	82		MOV MOVX	DPL, IFETCH A. @DPTR	; Load input buffer pointer ; Read character from FIFO	
24 0228	E0		• • • • • • • • • • • • • • • • • • • •		CLR	A, @DPTR C	; Flas everythins ok	
25 0229	22				RET			
						نو		
						ý		
						······j		

1					PCOF -	Place character in o	utput FIFO	
2 3				; ;	*PCOF*	places a single char	acter into the output FIFO and	
4				;	checks	for online and half	duplex.	
		• • • • • • • • • • • • • • • • • • • •						
7					ENTRY	(A) = character		
					EXIT	(A) = character	m in FIFO, clear otherwise	
10 11				; ;		'U' = set 1+ no roo	m in FIFU, clear otherwise	
12 13				,	ÜŚĘŚ	A, B, DPTR		
14								
.15 022A .16 022D	20 80	2E 9B	02	PCOF:	JB MĖ	ONLINE, PO1	; IF offline THEN ; place on input and return	
17			65		JB		; IF in half duplex mode THEN	
18 022F 19 0232	20 31	2F CA	02	P01:	ACALL	FULLDPLX, PO2 PCIF	; if in half duplex mode imen ; place on input also	
19 0232 20 21 0234		FO		P02:	MOV		; Save temporary	
727 0236	F5 É5	òF			MoV	B, A A, OSTORE		
. 23. 0238 . 24. 023 8	B4 	F7	04		CJNE SETB	A, #LOW (OLEN), POS	<pre>3</pre>	
25 0230	<u>E5</u>	F0			MOV RÉT	А, В	; restore char and return	
26 023E 27	22							
28 023F 29 0240	C3	on	=7	P03:	CLR MOV	C DPH, #HIGH (OBUF)	; ELSE clear 'C' ; set up the output	
30 0243	75 85	83 	57 82		MÖV	DPL, OSTORE	; buffer pointer	
.31 0246 .32 0248	E5	F0			MOV MOVX	A, B @DPTR, A	; restore character ; write character to buffer	
33 0249	05 22	OF			INC RET	OSTORE	; bump store and return	
13411024811	22				REI			
	• • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •					
• • • • • • • • • • • • • • • • • • • •								
	• • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •						
		•••••						
• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •						
								• • • • • • • •

Z29ROM; MICROBENCH 80 *** Z-29 COMPUTER TER	051 CROSS ASSEMB MINAL FIRMWARE	RLER (V1)-218 24 VERSION 1.03 ***	4-FEB-83 15 *	:30:06 PAGE 29	
				fetch character from ou	tput FIFO
2 3 4 5		, , ,	*FCOF*	fetches a character fro ailable.	m the output FIFO if any
6 7		5	ENTRY	none	
8 9 10	• • • • • • • • • • • • • • • • • • • •		EXIT	(A) = character if ava 'C' = set if no charac	ilable ters, clear otherwise
11 12 13		· · · · · · · · · · · · · · · · · · ·	USES	A, B, DPTR	
14 15 024C E5 16 024E B5	OF 02	FCOF:	MOV CUNE	A, OSTORE A, OFETCH, FOI	; IF store = fetch THEN
16 024E 85 17 0251 D3 18 0252 22			SETB RET	C	; set 'C' and return
19 20 0253 75	83 57	F01:	MOV	DPH, #HIGH (OBUF) DPL, OFETCH	A Just outside buffer pointer
21 0256 85 22 0259 E0 23 025A F5	10 82 F0		MOV MÖVX MOV	A, edetr B, A	; Load output buffer pointer ; Get character from buffer ; Save temporary
24 025C C2 25 025E 05	AF 10 OF		CLR INC MOV	EA OFETCH A, OSTORE	; Lock out entries ; Bump fetch pointer
27 28 0262 B5	10 06		CUNE	A, OFETCH, FOR	; IF store = fetch THEN
29 0265 75	10 F0		MOV	OFETCH, #LOW (OBUF)	; reset the values of
30 0268 75	of fo		MOV	OSTORE, #LOW (OBUF)	; fetch and store
30 0268 75 31 32 0268 21	OF FO	FOR:			; fetch and store ; Return (saves ROM)
31	OF FO	FOR:	MOV	OSTORE, #LOW (OBUF)	; fetch and store
31	OF FO	FOR:	MOV	OSTORE, #LOW (OBUF)	; fetch and store
31	OF FO	FOR:	MOV	OSTORE, #LOW (OBUF)	; fetch and store
31	OF FO	FOR:	MOV	OSTORE, #LOW (OBUF)	; fetch and store
31	OF FO	FOR:	MOV	OSTORE, #LOW (OBUF)	; fetch and store
31	OF FO	FOR	MOV	OSTORE, #LOW (OBUF)	; fetch and store
31	OF FO	FOR	MOV	OSTORE, #LOW (OBUF)	; fetch and store
31	OF FO	FOR	MOV	OSTORE, #LOW (OBUF)	; fetch and store
31	OF FO	FORE	MOV	OSTORE, #LOW (OBUF)	; fetch and store
31	OF FO	FORE	MOV	OSTORE, #LOW (OBUF)	; fetch and store
31	OF FO	FOR	MOV	OSTORE, #LOW (OBUF)	; fetch and store

.1					PCKB -	Place character of	it to the keyboard
.3 .3					*PCKB*	outputs bytes to t	he keyboard using a special handshake.
. 5		• • • • • • • • • • • • • • • • • • • •			embergera e	,,,,,	·······
. 7				; ;	ENTRY	(A) = byte to out	Put
8 9				;	EXIT	none	
.0 .1	• • • • • • • • • • • • • • • • • • • •			,	USES	A, B, WORK	
2 3 026D	12	102A		РСКВ:	LCALL	R2_KY2	; Keyboard hook
4 5 0270	30	91	oc		JNB	KBIN, PK1	; IF data THEN
6 0273	····čö····	ÉÖ			PUSH	ACC	; save byte
7 0275	51	B6			ACALL	FCKB	; fetch char from keyboard
8 0277	40	02			JĊ	PKO	; IF none now THEN exit
9.0279	51	EE			ACALL	KBCP	; ELSE process character
0 027B	DO	ËÖ		PKO:	POP	ACC	
1 027D 2	80	EE			SJMP	PCKB	; and try asain
2 027F	02	AF 90		PK1:	CLR	EA	; Disable interrupts
4 0281	C2 	90			CLR	KBOUT	Request keyboard attention
5 0283 6	75	F0	08		MOV	B, #8	; Initialize index for 8 bits
6 7 0286	30	91	06		JNB	KBIN, PK1A	; IF contention THEN
8 0289	30 D2	91 AF			SETB	EA	; enable interrupts
9 028B	D2				SETB	KBOUT	; wipe out request
0 0280	80	90 DE			SJMP	PCKB	; try again
1 21028F11 3	p.:	AF		PK1A:	SETB	EA	; Enable interrupts
3 4110291111	i3	• • • • • • • • • • • • • • • • • • • •		PKL:	RRC	····A······	; REPEAT rotate next bit in
5 0292	7C	00			MOV	WORK, #O	; init count for safety check
6 0294	26	9i	oe	PK2:	JB	KBIN, PK3	REPEAT wait for request
7 0297	51	B5			ACALL	PKDLY	; delay some time
8 0299 -	51	B5			ACALL	PKDLY	
9 029B	51	B5			ACALL	PKDLY	
0 029B	51	B5			ACALL	PKDLY	
1 029F	51	B5			ACALL	PKDLY	
2102A111	DC	F1			DUNZ	WORK, PK2	; UNTIL check is done
3 02A3 4	80	OE			SJMP	PKR	
5 02A5	92	90		PK3:	MOV	KBOUT, C	; output data
6 02A7	7C · · ·	95			MOV	WORK, #0	; init count for safety check
7 02A9	30	91	04	PK4:	JNB	KBIN, PK5	; REPEAT wait for signal
81102AC	ĎĊ	FB			DUNZ	WORK, PK4	; UNTIL safety check done
9 02AE	80	03			SJMP	PKR	
0	T)E	-	T) C	DIZE.	gr. 16.170	D D1/1	
1 02B0 2	D5	F0	DE	PK5:	DUNZ	B, PKL	; UNTIL all 8 bits are done
2 3 02 B 3	D2	90		PKR:	SETB	VDOUT	• Dankara akakus
3 02B3 4 02B5	22	90		PKDLY:	RET	KBOUT	; Restore status

					FCKB -	fetch character fro	m keyboard
				7 5	*FCKB* handsh		om the keyboard using a special
				, ,	ENTRY	none	
				; ; ;	EXIT	(A) = character if	available aracter, clear otherwise
 !					USES	A, B, DPTR	
02B6	12	1027		FCKB:	LCALL	R2_KY1	; Keyboard hook
0289 0280 0280	20 D3 22	91	02		JB SETB RET	KBIN, FK1 C	; IF no data THEN ; set carry and return
O2BE		39	00	FK1:	MOV	AUTOCNT, #0	
0201'''	75 D2	24			SETB	V\$PF	; Togale screen saver
0203 020 4	E4 75	FO	08		CLR MOV	A B, #8	; Init receiving byte ; Init index counter
7 0207	C2	90 E1		FKL:	CLR ACALL	KBOUT FKDLY	; REPEAT request data ; delay 200 usec
0209 020B	51 51 	E1 E1 91			ACALL	FKDLY	; set data bit
02CD 02CF	D2	90			MOV SETB	C, KBIN KBOUT	; data received ; place data into word
02Di 8 02D3	92 D5	E7 F0	02		MOV DJNZ	ACC.7, C B, FK2 FKR	; IF all 8 bits THEN
1 02D6 5	80	os			SJMP		; exit out
02D8 7 02D9	03 51	E1		FK2:	RR ACALL	A FKDLY	; rotate to next bit ; delay 100 usec
02DB	go	EA			SJMP	FKL	; UNTIL forever
) 02DD	5i	E1	• • • • • • • • • • • • • • • • • • • •	FKR:	ACALL	FKDLY	; Delay last time to let clear ; Clear 'C' for ok
. 02DF 2 02E0	Ç3 22				CLR RET	<u>C</u>	y oreat of tot or
} }		• • • • • • • • • • • • • • • • • • • •			Delay	routine	
02E1 02E4	75 D5	83 83	1E FD	FKDLY: FKDLY1:	MOV ZNLQ	ДРН, #30 ДРН, FKDLY1	; Delay of 100 usec
02E7	22				RET		

1. 2 3 4 5. 6		• • • • • • • • • • • • • • • • • • • •			IVD			
4 5 6				• • • • • • • • • • • • • • • • • • •	409.7	look.at.kemboand	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
			••••••		*LKB* to sen	looks at keyboard. d to the host	to.see.if.it.has.anv.data.	it.wants
	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	••••••		ENTRY	лоле.		
					EX.IT	(C) = set if no	character, clear otherwise	
		• • • • • • • • • • • • • • • • • • • •		, , , , , , , , , , , , , , , , , , ,		none		
12	02E8 C:							
15 16	02E9 20 02EC D:	?	01	LKB:	CLR JB SETR	C KBIN, LKBR C	; IF no data THEN; set carry and :	
17	02ED 2:	?		LKBR:	RET		, set carry and I	eturn
		• • • • • • • • • • • • • • • • • • • •			***************************************			
	• • • • • • • • • • • • • • • •	•••••	•••••	• • • • • • • • • • • • • • • • • • • •	*********			•••••
	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	•••••••	• • • • • • • • • • • • • • • • • • • •			
		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	••••••	• • • • • • • • • • • • • • • • • • • •			
						• • • • • • • • • • • • • • • • • • • •		
					••••			
						• • • • • • • • • • • • • • • • • • • •		•••••
	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	•••••	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	•••••	
	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	,	•••••	•••••

			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	•••••	
	• • • • • • • • • • • • • • • • • • • •							
						•••••	••••••	•••••
• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	**************	• • • • • • • • • • • • • • • • • • • •	•••••	• • • • • • • • • • • • • • • • • • • •	•••••	••••••	••••••
		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			••••••	
	• • • • • • • • • • • • • • • • • • • •					•••••		
	• • • • • • • • • • • • • • • • • • • •					• , • • • • • • • • • • • • • • • • • •	•••••	
	• • • • • • • • • • • • • • • •		******					
						•••••		•••••••••••••••••
• • • • • • • • • • • • • • • • • • • •		•••••••	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			
• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	•••••	

				; ;	KBCP -	keyboard character P	rocessor
				*		proceese any charac	ters received from the keyboard.
						FI OCESSES and chaire.	***.#(
	· • • • • • • • • • • • • • • • • • • •				ENTRY	(A) = byte from key	board
					EXIT	none	
				5	USES	all	
02EE	20	F7	03	KBCP:	JB	ACC.7, KPO	; IF ASCII character THEN
02F1	41	E7 2A			AJMP	PCOF	; output char and return
02F3	22			KPR:	RET		
02F4	A2	E6		KPO:	MOV	C, ACC.6	
02F6	92	1Ċ			MOV MOV	SHIFTF, C C, ACC.5	; Save shift value
.02F8 02FA	A2 92	E5			MOV	CONTLE, C	; Save control value
02FA 02FC	54	9F			ANL	A, #10011111B	: Mask off shift and control
	30	1É	02		JNB	ACCESSF, KP1	; IF access in progress THEN
0301	61	cp			AJMP	KPA	; do a lons jume
				.	No acc	ess in progress	
						B, #KBDTL	; (B) = table length
0303	75 20	FO 044B	09	KP1:	MOV MOV	DPTR, #KBDT	; (DPTR) = keyboard data table
0306	90 31	ЗАТЬ ЗА			ACALL	STAB	; Search the table
озов	40	02			JC	KP2	; IF found THEN
					Comman	d from keyboard	
· 				;			
OBOD	E4				CLR	Α	
. 030E	73				JMP	@A+DPTR	; Jump to routine
!					Cursor	keys	
					MOV	TEMP, A	
030F	FD <u>5</u> 5	OF		KP2:	MOV	A, OSTORE	; IF output FIFO not empty THEN
0312	<u>5</u> 7	10	DE		CUNE	A, OFETCH, KPR	; loose character and return
0315	ED				YQM	A, TEMP	
, 0316	75	F0	07		MOV	B, #KBCRTL	; (B) = table length
0319	90	0466			MOV	DPTR, #KBCRT	; (DPTR) = kb cursor table
0310	31	3A 2F			ACALL. JC	STAB KP6	; Search the table ; IF found THEN
031E	40	2F 42	no		JB JB	ADM3, KPR	; IF ADM 3 mode THEN exit
. 0320 2 0323	29 20	114 43	po CD		JB	HAZ, KPR	; IF Haz mode THEN exit
: 0323 3.0326		82			PU\$H	DPL	;save values
0328	CO	83			PUSH	DPH	
5. 932 A	7.4	1,B			MQV	A, #ESC	; output escape
0320	91 30	14 40	04		ACALL JNB	KPCOF ANSI, KP3	; output escape ; IF in ANSI mode THEN

.....

0335 91 14	0390 0393	B4 74 91	D7 1B 14	21		CJNE MOV	A, #KB_7 + KB_SHFT, A, #ESC	KP12
0333 91 14	038D	20	42 43 	DE DB 21	KP8:	JB JB	ADM3, KP16A HAZ, KP16A	; IF ADM 3 mode THEN ordinary ; IF Haz mode THEN ordinary
0333 91 14	γ.αφχ							
0335	• • • • • • • •		• • • • • • • • • • • • • • • • • • • •				mode hav	
0335					••••••		· · · · i ː · · · · · · · · · · · · · ·	
0335 P0 E0 KP3: PDP ACC 1 1F eval ker AND 0335 P0 E0 KP3: PDP ACC 1 1F eval ker AND 0335 P0 E0 KP3: PDP ACC 1 1F eval ker AND 0336 P0 E0 KP3: PDP ACC 1 1F eval ker AND 0336 P0 E0 KP3: PDP ACC 1 1F eval ker AND 0336 P0 E0 KP3: PDP ACC 1 1F in ANSI mode THEN 0336 P0 E0 KP3: PDP ACC 1 1F in ANSI mode THEN 0336 P0 E0 KP3: PDP ACC 1 1F in ANSI mode THEN 0336 PD E0 KP3: PDP ACC 1 1F in ANSI mode THEN 0341 P1 14 ACALL KPCOF 1 0 utput 2' Several MOV DPL A 1 ELSE in ZDS mode 0342 P0 E0 KP3: PDP ACC 1 1 eval ker Ang. A stack 0344 P0 E0 KP3: PDP ACC 1 1 eval ker Ang. A stack 0344 P0 E0 KP3: PDP ACC 1 1 eval ker Ang. A stack 0344 P0 E0 KP3: PDP ACC 1 1 eval ker Ang. A stack 0344 P0 E0 KP3: PDP ACC 1 1 eval ker Ang. A stack 0344 P0 E0 KP3: PDP ACC 1 1 eval ker Ang. A stack 0344 P0 E0 KP3: PDP ACC 1 1 eval ker Ang. A stack 0344 P0 E0 KP3: PDP ACC 1 1 eval ker Ang. A stack 0344 P0 E0 KP3: PDP ACC 1 1 eval ker Ang. A stack 0344 P0 E0 KP3: PDP ACC 1 1 eval ker Ang. A stack 0344 P0 E0 KP3: PDP ACC 1 1 eval ker Ang. A stack 0344 P0 E0 KP3: PDP ACC 1 1 eval ker Ang. A stack 0344 P0 E0 KP3: PDP ACC 1 1 eval ker Ang. A stack 0344 P0 E0 KP3: PDP ACC 1 1 eval ker Ang. A stack 0344 P0 E0 KP3: PDP ACC 1 1 eval ker Ang. A stack 0344 P0 E0 KP3: PDP ACC 1 1 eval ker Ang. A stack 0344 P0 E0 KP3: PDP ACC 1 1 eval ker Ang. A stack 0344 P0 E0 KP3: PDP ACC 1 1 eval ker E0 KP3: PDP ACC 1 eval ker E0 K	0388							: Output final char and actuar
0335 91 14 ACALL PPOP OF OUTPUT TO OUTPUT TO 0335 DO EQ. KP3: POP ACC. 1 OUTPUT TO 0335 DO EQ. KP3: POP ACC. 1 IF erase key AND 12 UZ KP4 1 IF erase key AND 12 UZ KP4 1 IF erase key AND 1339 30 15, OF JNB SHIFTF, KP4 1 IF erase key AND 1339 30 15, OF JNB SHIFTF, KP4 1 IF erase key AND 1339 30 15, OF JNB SHIFTF, KP4 1 IF erase key AND 1339 30 15, OF JNB ANSI. KP5 1 IF in ANSI mode THEN 10336 30 40 06 JNB ANSI. KP5 1 IF in ANSI mode THEN 10336 13 14 ACALL KPCOF 1 OUTPUT ZZ 14 14 ACALL KPCOF 1 OUTPUT ZZ 15 MODE 14 ACALL KPCOF 1 OUTPUT ZZ 15 MODE 15 MOD DPL. A 1 ELSE in ZDS mode 15 MOD DPL. A 1 ELSE in ZDS mode 15 MOD DPL. A 1 ELSE in ZDS mode 15 MOD DPL. A 1 ELSE in ZDS mode 15 MOD DPL. A 1 ELSE in ZDS mode 15 MOD DPL. A 1 ELSE in ZDS mode 15 MOD DPL. A 1 ELSE in ZDS mode 15 MOD DPL. A 1 ELSE in ZDS mode 15 MOD DPL. A 1 ELSE in ZDS mode 15 MOD DPL. A 1 ELSE in ZDS mode 15 MOD DPL. A 2 ELSE in ZDS mode 15 MOD DPL. A 2 ELSE in ZDS mode 15 MOD DPL. A 2 ELSE in ZDS mode 15 MOD DPL. A 2 ELSE in ZDS mode 15 MOD DPL. A 2 ELSE in ZDS mode 15 MOD DPL. A 2 ELSE in ZDS mode 15 MOD DPL. A 2 ELSE in ZDS mode 15 MOD DPL. A 2 ELSE in ZDS mode 15 MOD DPL. A 2 ELSE in ZDS mode 15 MOD DPL. A 2 ELSE in ZDS mode 15 MOD DPL. A 2 ELSE in ZDS mode 15 MOD DPL. A 2 ELSE in ZDS mode 15 MOD DPL. A 2 ELSE in ZDS mode 15 MOD DPL. A 2 ELSE in ZDS mode 15 MOD DPL. A 2 ELSE in ZDS mode 15 MOD DPL. A 2 ELSE in ZDS MOD DPL. A 2 ELSE I	0386	····á			KF11:			; output /?/
0335 PJ 14 ACALL PPOP OF 0 output T 0 0335 PJ 16 PJ 16 PPOP ACC	0384				VD11.			; ELSE IF in ZDS mode THEN
14				02			ANSI, KP11	
14			• • • • • • • • • • • • • • • • • • • •					; IF in ANSI mode THEN
14		21						
0335 91 14 0335 90 E0 KP3: P9P ACC ; output 'I' 0337 60 12 0337 60 12 032	0379							
0335 91 14 0335 90 E0 KP3: P9P ACC ; output 'I' 0337 60 12 0337 60 12 032	9377		82					, in max mode IMEN ordinary
0335 91 14 ACALL KPCOF ; output '[' 0335 D0 E0 KP3: PDP ACC ; IF erase key AND 0337 40 12 JZ KP4 ; IF erase key AND 0337 40 12 JZ KP4 ; IF erase key AND 0338 30 40 06 JNB SHLTIF, KP4 ; IF shift key down. THEN 0330 30 40 06 JNB ANSI, KP5 ; IF in ANSI mode THEN 0337 74, 32 MDV A, #'.2' 0341 91 14 ACALL KPCOF ; output '2' 0343 30 06 SJMP KP4 0347 D0 E0 PDP ACC ; replace char on stack 0349 C0 82 PUSH DPL ; replace char on stack 0349 C0 82 PUSH DPL ; to be used below 0340 81 14 ACALL KPCOF ; Output next char and return 0340 81 14 ACAMP KPCOF ; Output next char and return 0340 81 14 ACAMP KPCOF ; Output next char and return 0351 30 30 01 JNB KPADSHTF, KP7 ; IF keypad is shifted THEN 0351 30 30 01 JNB KPADSHTF, KP7 ; IF keypad is shifted THEN 0351 30 30 01 JNB KPADSHTF, KP7 ; IF keypad is shifted THEN 0352 92 IC KP6: MOV SHIFTF C ; Save final result 0353 93 CPL KP7: MOV SHIFTF C ; Save final result 0355 92 IC KP7: MOV SHIFTF C ; Save final result 0355 92 IC KP7: MOV SHIFTF C ; Save final result 0357 72 E6 MOV ACC, 6, C ; slace into data word also one of the stack of the stac	0374		43	F4				J. Ar Mum 3 mode THEN ordinary
03353 91 14 ACALL KPCOF ; output [7] 03355 D0 E0 KP3: POP ACC ; output [7] 03357 60 12 JZ KP4 ; IF erase kev AND 03373 30 1C OF JNB SHLETF, KP4 ; IF erase kev AND 03387 30 1C OF JNB SHLETF, KP4 ; IF shift kex down THEN 03387 30 40 06 JNB ANSI, KP5 ; IF in ANSI mode THEN 03387 74 32 MQV A, #;2' 0341 91 14 ACALL KPCOF ; output [2] 03437 39 0, 06 SJMP KP4 0343 39 0, 06 SJMP KP4 0347 D0 E0 KP5: MOV DPL, A ELSE in ZDS mode 0345 F5 82 KP5: MOV DPL, A ELSE in ZDS mode 0347 D0 E0 POP ACC ; replace, char, an, stack 0349 C0 82 PUSH DPL ; to be used below 0349 D0 E0 KP4: POP ACC ; to be used below 0340 81 14 AJMP KPCOF ; output next char and return 0350 30 30 01 JNB KPADSHTE, KP7 ; IF kevpad is shifted THEN 0351 30 30 30 01 JNB KPADSHTE, KP7 ; IF kevpad is shifted THEN 0353 92 IC KP7: MOV SHIFTE, C ; save final result 0355 92 IC KP7: MOV SHIFTE, C ; save final result 0357 72 E6 MOV SHIFTE, C ; save final result 0357 72 E6 MOV DPTR, MFBPDT (B) etable length 0356 75 PO 047B MOV DPTR, MFBPDT (B) etable length 0357 31 3A ACALL STAB ; search the table 0358 30 SE SJMP KPR ; IF kevpad unshifted THEN 0368 20 31 06 JB KPADALTE, KP9 ; IF kevpad unshifted THEN 0368 20 31 06 JB KPADALTE, KP9 ; IF kevpad unshifted THEN 0368 E5 S3 KP16A: MOV A DPH.	0371	20	42	F7	KP9:	. IR	ABMR. KD14A	A TE ADM O TURN
03353 91 14 ACALL KPCOF ; output [7] 03355 D0 E0 KP3: POP ACC ; output [7] 03357 60 12 JZ KP4 ; IF erase kev AND 03373 30 1C OF JNB SHLETF, KP4 ; IF erase kev AND 03387 30 1C OF JNB SHLETF, KP4 ; IF shift kex down THEN 03387 30 40 06 JNB ANSI, KP5 ; IF in ANSI mode THEN 03387 74 32 MQV A, #;2' 0341 91 14 ACALL KPCOF ; output [2] 03437 39 0, 06 SJMP KP4 0343 39 0, 06 SJMP KP4 0347 D0 E0 KP5: MOV DPL, A ELSE in ZDS mode 0345 F5 82 KP5: MOV DPL, A ELSE in ZDS mode 0347 D0 E0 POP ACC ; replace, char, an, stack 0349 C0 82 PUSH DPL ; to be used below 0349 D0 E0 KP4: POP ACC ; to be used below 0340 81 14 AJMP KPCOF ; output next char and return 0350 30 30 01 JNB KPADSHTE, KP7 ; IF kevpad is shifted THEN 0351 30 30 30 01 JNB KPADSHTE, KP7 ; IF kevpad is shifted THEN 0353 92 IC KP7: MOV SHIFTE, C ; save final result 0355 92 IC KP7: MOV SHIFTE, C ; save final result 0357 72 E6 MOV SHIFTE, C ; save final result 0357 72 E6 MOV DPTR, MFBPDT (B) etable length 0356 75 PO 047B MOV DPTR, MFBPDT (B) etable length 0357 31 3A ACALL STAB ; search the table 0358 30 SE SJMP KPR ; IF kevpad unshifted THEN 0368 20 31 06 JB KPADALTE, KP9 ; IF kevpad unshifted THEN 0368 20 31 06 JB KPADALTE, KP9 ; IF kevpad unshifted THEN 0368 E5 S3 KP16A: MOV A DPH.		· · · · · · · · · · · ·	· · · · · · · * · · · · · · · ·	• • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		NER	
03323 91 14 ACALL KPCOF ; output [7] 0335 D0 E0 KP3: POP ACC ; output [7] 0335 D0 E0 KP3: POP ACC ; IF erase key AND 0336 30 12 OF JNB SHLETF, KP4 ; IF erase key AND 0339 30 10 O6 JNB SHLETF, KP5 ; IF in ANSI mode THEN 0336 30 40 06 JNB ANSI, KP5 ; IF in ANSI mode THEN 0337 74 32 MOV A, #\frac{1}{2}\$ MOV A, #\frac{1}{2}\$ MOV BPL A ; ELSE in ZDS mode 0345 F5 82 KP5: MOV DPL A ; ELSE in ZDS mode 0347 D0 E0 KP4: POP ACC ; replace char an stack 0348 D0 E0 KP4: POP ACC ; replace char an stack 0348 D0 E0 KP4: POP ACC ; output rext char and return 0340 81 14 AJMP KPCOF ; output next char and return 0341 MOV DPL A ; ELSE in ZDS mode 0342 C 82 PUSH DPL ; to be used below 0343 S1 14 AJMP KPCOF ; output next char and return 0348 D0 E0 KP4: POP ACC ; save final result 0351 30 30 01 JNB KPADSHTF, KP7 ; IF keypad is shifted THEN 0351 30 30 01 JNB KPADSHTF, KP7 ; IF keypad is shifted THEN 0353 92 IC KP6: MOV SHIFTF, C ; save final result 0355 92 IC KP7: MOV SHIFTF, C ; save final result 0355 92 IC KP7: MOV SHIFTF, C ; save final result 0355 93 IS A ACALL STAB SHEPDT ; (DPTR) = keypad table 0356 30 SE SJMP KPR ; exit out 0368 SE SJMP KPR ; exit out 0368 SE SJMP KPR ; Extract or in the strength of the solid								; output ordinary character
0333 91 14 ACALL KPCOF ; output 'I' 0335 D0 E0 KP3: PGP ACC 0337 60 12 JZ KP4 ; IF erase key AND 0338 30 1C OF JNB SHIFTF, KP4 ; IF, shift key down THEN 033C 30 40 06 JNB ANSI, KP5 ; IF in ANSI mode THEN 033F 74 32 MOV, A, #'2' ; IF in ANSI mode THEN 0341 91 14 ACALL KPCOF ; output '2' 0343 80 06 SJMP KP4 0347 D0 E0 KP5: MOV DPL, A ; ELSE in ZDS mode 0347 D0 E0 POP ACC ; replace char on stack 0349 C0 82 PUSH DPL ; to be used below 034B D0 E0 KP4: POP ACC ; replace char on stack 034B S1 14 AJMP KPCOF ; Output next char and return 034B D0 E0 KP4: MOV C, SHIFTF 0351 30 30 01 JNB KPADSHTF, KP7 ; IF keypad is shifted THEN 0354 B3 CPL C 0355 2 IC KP7: MOV SHIFTF, C ; save final result 0357 92 E6 MOV SHIFTF, C ; save final result 0357 92 E6 MOV DPT, KREPTT ; (B) = table length 0356 90 047B MOV DPT, KREPTT ; (B) = table length 0363 80 SE SJMP KPR ; IF keypad unshifted THEN 0365 20 IC 22 KP16: JB SHIFTF, KP8 ; IF keypad is ordinary THEN 0366 20 31 06 JB KPADALTF, KP9 ; IF keypad is ordinary THEN	036D			• • • • • • • • • • • • • • • • • • • •	NF.+9.H*			
0333 91 14 ACALL KPCOF ; output [7] 0335 D0 E0 KP3: POP ACC ; if erase key AND 0337 60 12 JZ KP4 ; IF erase key AND 0339 30 16 OF JNB SHIFTF, KP4 ; IF, shift, key, down, THEN 0330 30 40 06 JNB ANSI, KP5 ; IF in ANSI mode THEN 0337 74 32 MQV A, #(2') 0341 91 14 ACALL KPCOF ; output [2'] 0343 30 06 SJMP KP4 03443 90 06 SJMP KP4 0345 F5 82 KP5: MGV DPL, A ; ELSE in ZDS mode 0347 D0 E0 F0 KP4: POP ACC ; replace char an stack 0349 C0 82 PUSH DPL ; to be used below 0340 81 14 AJMP KPCOF ; Output next char and return 0341 81 14 AJMP KPCOF ; Output next char and return 0341 83 O O1 JNB KPADSHTF, KP7 ; IF keypad is shifted THEN 0354 83 O O1 JNB KPADSHTF, KP7 ; IF keypad is shifted THEN 0355 92 IC KP7: MGV SHIFTF CP ; save final result 0357 75 F0 1C MGV SHIFTF, C ; save final result 0358 MGV B #KBPDTL ; B) = keypad is hift dynation 0357 75 F0 1C MGV B #KBPDTL ; B) = cable length 0363 30 8E JMP KPR ; exit out 0365 20 IC 22 KP16: JB SHIFTF, KP8 ; IF keypad unshifted THEN	036B			00	VD14A.			; IF keypad is ordinary THEN
0333 91 14 ACALL KPCOF	0368		31		•••••		MDADAL TE MES	
0333 91 14 ACALL KPCOF ; output 'C' 0335 DO EQ KP3: POP ACC 0337 60 12 JZ KP4 ; IF erase key AND 0337 30 1C OF JNB. SHIFTF, KP4 ; IF shift key down THEN 0330 30 40 06 JNB ANSI, KP5 ; IF in ANSI mode THEN 0331 91 14 ACALL KPCOF ; output 'C' 0341 91 14 ACALL KPCOF ; output 'C' 0343 90 06 SJMP KP4 0345 F5 82 KP5: MOV DPL, A ; ELSE in ZDS mode 0347 DO EQ PDP ACC ; replace char on stack 0349 CO 82 PUSH DPL ; to be used below 034B DO EQ KP4: POP ACC ; Output next char and return 034B DO EQ KP4: MOV C, SHIFTF 0351 30 30 01 JNB KPADSHTF, KP7 ; IF keypad is shifted THEN 0355 92 IC KP7: MOV SHIFTF, C ; save final result 0355 92 IC KP7: MOV SHIFTF, C ; save final result 0357 92 E6 MOV SHIFTF, C ; save final result 0357 92 E6 MOV SHIFTF, C ; save final result 0357 92 E6 MOV BH KPADSHTF, KP7 ; IF keypad is shifted THEN 0357 92 E6 MOV SHIFTF, C ; save final result 0357 97 F0 IC MOV SHIFTF, C ; save final result 0358 99 T5 F0 IC MOV BH KREPDTL ; (B) = table lensth 0358 90 SE SJMP KPR ; exit out	0365		10	22	KP16:	JB	SHIFTF, KP8	; IF keypad unshifted THEN
0333 91 14 ACALL KPCOF ; output [7] 0335 D0 E0 KP3: POP ACC 0337 60 12 JZ KP4 ; IF erase key AND 03387 30 1C OF JNB SHIFTE, KP4 ; IF shift key down THEN 0330 30 40 06 JNB ANSI, KP5 ; IF in ANSI mode THEN 03317 74, 32 MRY A, #(2) 0341 91 14 ACALL KPCOF ; output [2] 0343 90 06 SJMP KP4 0345 F5 82 KP5: MOV DPL, A ; ELSE in ZDS mode 0347 D0 E0 POP ACC ; replace char on stack 0349 C0 82 PUSH DPL ; to be used below 034B D0 E0 KP4: POP ACC 035B 30 30 01 JNB KPADSHTF, KP7 ; IF keypad is shifted THEN 035C 90 LC KP7: MOV SHIFTF C ; invert. shift, function 035C 91 C MOV SHIFTF C ; save final result 035C 92 LC KP7: MOV SHIFTF C ; save final result 035C 90 047B MOV DPTR, #KBPDT ; (DPTR) = keypad table 036C 90 047B MOV DPTR, #KBPDT ; (DPTR) = keypad table 036C 90 UNC KP16 ; IF not found THEN						SJMP	KPR	; exit out
0333 91 14			<u>94</u>	• • • • • • • • • • •				; IF not found THEN
0333 91 14 ACALL KPCOF ; output [] 0335 DO EO KP3: POP ACC 0337 30 1C OF JNB SHIFTF, KP4 ; IF erase key AND 0330 30 1C OF JNB ANSI, KP5 ; IF in ANSI mode THEN 0331 74 32 MOV A, # (2) 0341 91 14 ACALL KPCOF ; output [2] 0343 80 06 SJMP KP4 0345 F5 82 KP5: MOV DPL, A ; ELSE in ZDS mode 0347 DO EO POP ACC 0348 DO EO KP4: POP ACC 0349 CO 82 PUSH DPL ; to be used below 0340 81 14 AJMP KPCOF ; Output next char and return Numeric keypad key 0351 30 30 01 JNB KPADSHTF, KP7 ; IF keypad is shifted THEN 0355 92 1C KP7: MOV SHIFTF 0357 92 E6 MOV SHIFTF 0350 C								; search the table
0333 91 14 ACALL KPCOF ; output '[' 0335 D0 E0 KP3: POP ACC 0337 60 12 0339 30 1C OF JNB SHIFTF, KP4 ; IF erase key AND 0330 30 40 06 JNB ANSI, KP5 ; IF in ANSI mode THEN 0331 91 14 ACALL KPCOF ; output '2' 0341 91 14 ACALL KPCOF ; output '2' 0345 P5 82 KP5: MOV DPL, A ; ELSE in ZDS mode 0347 D0 E0 POP ACC ; replace char on stack 0348 D0 E0 KP4: POP ACC 034B D0 E0 KP4: POP ACC 0355 92 IC KP6: MOV C, SHIFTF 0355 92 IC KP7: MOV SHIFTF, C ; invert. shift. function 0355 92 IC KP7: MOV SHIFTF, C ; save final result 0355 92 E6 MOV ACC. 6. C ; place into data word also	033U						DPTR, #KBPDT	; (DPTR) = keypad table
0333 91 14 ACALL KPCOF ; output '[' 9335 D0 E0 KP3: POP ACC 0337 60 12 0339 30 1C OF JNB SHIFTF, KP4 ; IF erase key AND 0330 30 40 06 JNB ANSI, KP5 ; IF in ANSI mode THEN 0331 91 14 ACALL KPCOF ; output '2' 0341 91 14 ACALL KPCOF ; output '2' 0343 90 06 SJMP KP4 0345 F5 82 KP5: MOV DPL, A ; ELSE in ZDS mode 0347 D0 E0 POP ACC ; replace char on stack 0349 C0 82 PUSH DPL ; to be used below 0349 C0 82 PUSH DPL ; to be used below 0340 81 14 AJMP KPCOF ; Output next char and return 0341 91 14 AJMP KPCOF ; if keypad is shifted THEN 0351 30 30 01 JNB KPADSHTF, KP7 ; IF keypad is shifted THEN 0353 B3 CPL C KP6: MOV C, SHIFTF 0351 30 30 01 JNB KPADSHTF, KP7 ; IF keypad is shifted THEN 0355 92 1C KP7: MOV SHIFTF, C ; save final result 0357 92 E6 MOV SHIFTF, C ; save final result 0357 92 E6 MOV SHIFTF, C ; save final result	0337			1C			B, #KBPDTL	; (B) = table length
0333 91 14 ACALL KPCOF ; output '[' 0335 DO EO KP3: POP ACC 0337 60 12 JZ KP4 ; IF erase key AND 0339 30 1C OF JNB SHIFTF, KP4 ; IF shift kex down THEN 0330 30 40 06 JNB ANSI, KP5 ; IF in ANSI mode THEN 0331 74 32 MOV A, #/2' 0341 91 14 ACALL KPCOF ; output '2' 0343 80 06 SJMP KP4 0345 F5 82 KP5: MOV DPL, A ; ELSE in ZDS mode 0347 DO EO POP ACC ; replace char so, stack 0348 DO EO KP4: POP ACC 0349 CO 82 0349 CO 82 0349 DO EO KP4: POP ACC 0340 AJMP KPCOF ; Output next char and return Numeric keypad key	0357	<u>?2</u>	<u>5</u> 6				ACC, 6, C	
0333 91 14	0355 0357				KP7:	MOV	SHIFTF, C	
0333 91 14 ACALL KPCOF ; output [, , ,			CPL	Ç	
0333 91 14			30	01		JNB	KPADSHTF, KP7	; IF keypad is shifted THEN
0333 91 14	034F	<u>A</u> 2	1Ç		KP6:	MOV	C, SHIFTF	
0333 91 14 ACALL KPCOF ; output '[' 0335 DQ EQ KP3: PQP ACC 0337 60 12 JZ KP4 ; IF erase key AND 0337 30 1C OF JNB SHIFTF, KP4 ; IF shift kex down THEN 033C 30 40 06 JNB ANSI, KP5 ; IF in ANSI mode THEN 033F 74 32 MQV A, #'2' Output '2' 0341 91 14 ACALL KPCOF ; output '2' 0343 80 Q6 SJMP KP4 Output '2' 0345 F5 82 KP5: MOV DPL, A ; ELSE in ZDS mode 0347 DO EO POP ACC ; replace char on stack 0349 CO 82 PUSH DPL ; to be used below 034B DO EO KP4: POP ACC ; Output next char and return			• • • • • • • • • • • • • • •		**************************************	Ndilist.	re. veilen Kei	
0333 91 14 ACALL KPCOF ; output '[' 0335 DO EO KP3: POP ACC 0337 60 12 JZ KP4 ; IF erase key AND 0337 30 1C OF JNB SHIFTF, KP4 ; IF shift key down THEN 033C 30 40 06 JNB ANSI, KP5 ; IF in ANSI mode THEN 033F 74 32 MQV A, #.22 Output '2' 0341 91 14 ACALL KPCOF ; output '2' 0343 30 06 SJMP KP4 0345 F5 82 KP5: MOV DPL, A ; ELSE in ZDS mode 0347 DO EO POP ACC ; replace char on stack 0349 CO 82 PUSH DPL ; to be used below 034B DO EO KP4: POP ACC ; Output next char and return					•	Ni ima	ic kayasa ker	
0333 91 14	• • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •					
0333 91 14 ACALL KPCOF ; output <[\infty \) 0335 DQ EQ KP3: PQP ACC 0337 60 12 JZ KP4 ; IF erase key AND 0339 30 1C OF JNB SHIFTF, KP4 ; IF, shift key down THEN 033C 30 40 06 JNB ANSI, KP5 ; IF in ANSI mode THEN 033F 74 32 MOV A, #/2/ 0341 91 14 ACALL KPCOF ; output <2 \(\) 0343 80 06 SJMP KP4 0345 F5 82 KP5: MOV DPL, A ; ELSE in ZDS mode 0347 DO EQ POP ACC ; replace char on stack 0348 DO EQ KP4: POP ACC	03 4 D	81	14			AJMP		; Output next char and return
0333 91 14 ACALL KPCOF ; output '[' 0335 D0 E0 KP3: POP ACC 0337 60 12 JZ KP4 ; IF erase key AND 0338 30 1C OF JNB SHIFTF, KP4 ; IF shift key down THEN 033C 30 40 06 JNB ANSI, KP5 ; IF in ANSI mode THEN 033F 74 32 MOV A, #'2' 0341 91 14 ACALL KPCOF ; output '2' 0343 90 06 SJMP KP4 0345 F5 82 KP5: MOV DPL, A ; ELSE in ZDS mode 0347 D0 E0 POP ACC ; replace char on stack 0349 C0 82 PUSH DPL ; to be used holosy	03 4B			· • • • • • • • • • • • • •	KP4:		ACC	, to be ased helom
0333 91 14 ACALL KPCOF ; output '[' 0335 D0 E0 KP3: POP ACC 0337 60 12 JZ KP4 ; IF erase key AND 0338 30 1C OF JNB SHIFTF, KP4 ; IF shift key down THEN 033C 30 40 06 JNB ANSI, KP5 ; IF in ANSI mode THEN 033F 74 32 MOV A, #'2' 0341 91 14 ACALL KPCOF ; output '2' 0343 90 06 SJMP KP4 0345 F5 82 KP5: MOV DPL, A ; ELSE in ZDS mode 0347 D0 E0 POP ACC	0349		82		• • • • • • • • • • • • • • • • • • • •		DPL	
0333 91 14 ACALL KPCOF ; output '[' 0335 D0 E0 KP3: POP ACC 0337 60 12 JZ KP4 ; IF erase key AND 0338 30 1C OF JNB SHIFTF, KP4 ; IF, shift key down THEN 033C 30 40 06 JNB ANSI, KP5 ; IF in ANSI mode THEN 033F 74 32 MOV A, #'2' 0341 91 14 ACALL KPCOF ; output '2' 0343 90 06 SJMP KP4	0347		E0				ACC	
0333 91 14 ACALL KPCOF ; output '[' 0335 D0 E0 KP3: POP ACC 0337 60 12 JZ KP4 ; IF erase key AND 0339 30 1C OF JNB SHIFTF, KP4 ; IF shift key down THEN 033C 30 40 06 JNB ANSI, KP5 ; IF in ANSI mode THEN 033F 74 32 MOV A, #'2' 0341 91 14 ACALL KPCOF ; output '2'	0345	F5			KP5:			
0333 91 14 ACALL KPCOF ; output (E	.0343							
0333 91 14 ACALL KPCOF ; output '[' 0335 D0 E0 KP3: POP ACC 0337 60 12 JZ KP4 ; IF erase key AND 0339 30 1C OF JNB SHIFTF, KP4 ; IF shift key down THEN 033C 30 40 06 JNB ANSI, KP5 ; IF in ANSI mode THEN	0341			• • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			
0333 91 14 ACALL KPCOF ; output '[' 0335 D0 E0 KP3: POP ACC 0337 60 12 JZ KP4 ; IF erase key AND 0339 30 1C OF JNB SHIFTF, KP4 ; IF shift key down THEN	.033F.		32	06				
0333 91 14 ACALL KPCOF ; output [9335 DO EO KP3: POP ACC 0337 60 12 JZ KP4 : IF erase key AND	0330			?Y			SHILLE, KP4	
0333 91 14 ACALL KPCOF ; output <[< 9335 DQ EQ KP3: PQP ACC ;				OE.				
O333 91 14 ACALL KPCOF : output </td <td></td> <td></td> <td></td> <td></td> <td>КРЗ:</td> <td></td> <td></td> <td></td>					КРЗ:			
2000 - A					1 2 80 20			; output '['
933174 5B MOU A. #474						МΩҮ	A>#1[.1	• • • • • • • • • • • • • • • • • • • •

C

	! =!> ! :=!>	Litindr-Lit	MANITY YE	RSION 1.03 *			
1 0297	20	40	09		JB	ANSI, KP13	; IF in ZDS mode THEN
1 0397 2 039A	20 74	4ö			MOV	A, #/@/	
3 0390	30	21	02		JNB	ICMODEF, KP14	; check if in IC mode
4.039F	74	21 4F			MOV	A, #282	
5 03A1	81	14		KP14:	AJMP	KPCOF	; output char and return
	• • • • • • • • •						. IT :- ANOT THEN
7 03A3	74	5B		KP13:	MOV	A, #1[1 KPCOF	; IF in ANSI mode THEN ; output 'C'
8 03A5	91	14			ACALL		; 00(P0(L
9 03A7	74	34			MOV ACALL	A, #′4′ KPCOF	; output /4/
10 03A9	91	14			MOV	A, #'h'	, 040,40
11 03AB	74	68 2i	öż		BNP	TCMODEF, KP15	; IF in IC mode THEN
12 03AD	30	6C	02		MOV	A, #111	,
13 03B0 14 03B2	81	14		KP15:	·····AJMP	KPCOF	; output char and return
14 0382	01	1.4		10. 10.			
·16·····	• • • • • • • • •						
17				;	Numeri	c keypad shifted	
18						TEMP A	
19 03B4	FD			KP12:	MOV	TEMP, A WORK, DPL	; Save second char
20 03B5	AC	82			MOV		, oave second char
21 0387	E5	83			MOV ACALL	A, DPH KPCOF	; Output first key
22 03B9	91	14			MOV	A, WORK	, 000100 12120 1101
23 03BB	EC	55			JŽ	KPR2	; IF second char not null THEN
24 03BC	60		0.0		JNB	ANSI, KP10	,
25 03BE	30 74	40 58	OA			A, #'t'	
26 0301	74 91	14			ACALL	KPCOF	; IF ANSI THEN output '['
27 03C3 28 03C5	? EC				MOV	A, WORK	•••••
29 0306	B4	4F	02		CUNE	A, #'N', KP10	; IF delete THEN ; change it for ANSI mode
30 0309	74	4E 50			MOV	A, #/P/	
31 03CB	81	14		KP10:	AJMP	KPCOF	; output second char & return
32							
33						······	
34				;	Access	code in progress	
. 35				; KPA:	CLR	ACCESSF	; Take out of access mode
36 03CD	c2	1E		NEM*	MOV	TEMP, A	; Save for a while
37 03CF 38 03D0	FD	F8			ANL	A, #11111000B	; Mask for keyboard ID number
38 03D0 39 03D2	B4	98 	06		CUNE	A, #KB_ID, KPB	; IF ID number THEN
.39.03D2 40.03D5	<u>F</u> 7				MOV	A, TEMP	; get original data
41 03D6	54	07			ANL	A, #00000111B	; mask for ID number
42 03D8	F5	1D			MOV	KBIDNUM, A	; save ID number
43 03DA	22				RET	,	
44							
.45						an kay	
46				; •	runct1	on key	
.47	<u>20</u>	42	35	, KPB:	JB	ADM3, KPR2	; IF ADM 3 mode OR
48 03DB 49 03DE	20	42 43	32	17s1 Aut #	JB	HAZ, KPR2	; IF Haz mode THEN exit
49V3DE 50	:						
51_03E1	ED				MOV	A, TEMP	; Get original character back
949954 52	· · · · · · · · · · · · ·						
53 03E2	75	FQ	09		MOV	B, #KBFNTL	; (B) = table length
54 03E5	90	04CF			MOV	DPTR, #KBFNT	; (DPTR) = function key table
.55.03E8	31	3A			ACALL.	STAB	; Search the table ; IF found THEN
56 03EA	40	27			JC	KPR2	; ir touna imen

	103EE	<u>CO</u>	83 1B			PUSH	DPH	
	2 03F0					MOV	A, #ESC	
• • • • • • • • • • • • • • • • • • • •	3.03F2 4 03F4	<u>91</u> 30	<u>14</u>	OB	• • • • • • • • • • • • • • • • • • • •	ACALL.	KPCOF	; output an ESC
	5 03F7	74	4F	OD		JNB MOV	ANSI, KPC A, #101	; IF in ANSI mode THEN
	6 03F9	91	14			ACALL	R	; output 'O'
	7. 03FB	DO.	E0			POP	ACC	, , , , , , , , , , , , , , , , , , , ,
	8 03FD	91	14			ACALL	KPCOF	; output next character
• • • • • • • • • • • • • • • • • • • •	9 03FF 10 0401	<u>D0</u>	<u>E</u> 0			POP RET	ACC	
	11					KEI		
	12 0402	DO	ĖÖ		KPC:	POP	ACC	; IF in ZDS mode THEN
	13 0404	DO	E0			POP	ACC	; get proper char
	14 0406	30	E7	ÓВ		JNB	ACC.7, KPCOF	f IF high bit set THEN
	15 0409 16 040B	C2 FD	E7			CLR	ACC.7	; clear hish bit
	17 040C		30			MOV MOV	TEMP, A	
• • • • • • • • • • • • • • • • • • • •	18 040E	74	14	• • • • • • • • • • • • • • • • • • • •		ACALL	. A, #′0′ KPCÓF	
	19 0410	ED				MOV	A, TEMP	; output 'O'
	20 0411	81	14			AJMP	KPCOF	; output next character
• • • • • • • • • • • • • • • • • • • •	.21 0413 .22	22			KPR2:	RET		
	.23 .24			• • • • • • • • • • • • • • • •		Check	76 12651 60626326 383	d execute proper routine
	25				;		i rocal ranction and	s execute brober contine
	26 0414	20	iD	02	KPCOF:	Jġ	CONTLE, KPC1	; IF control not pressed THEN
	27 0417	20 41	1D 2A	02	KPCOF:	JB AJMP	CONTLF, KPC1 PCOF	; IF control not pressed THEN ; output char and return
	.27 .0417 .28	41	2A	02		AJMP	PCOF	; output char and return
	27 0417	20 41 21		02	KPCOF:			; IF control not pressed THEN ; output char and return ; ELSE do local and return
	.27 .0417 .28	41	2A	02		AJMP	PCOF	; output char and return
	.27 .0417 .28	41	2A	02		AJMP	PCOF	; output char and return
	.27 .0417 .28	41	2A	02		AJMP	PCOF	; output char and return
	.27 .0417 .28	41	2A	02		AJMP	PCOF	; output char and return
	.27 .0417 .28	41	2A	02		AJMP	PCOF	; output char and return
	.27 .0417 .28	41	2A	02		AJMP	PCOF	; output char and return
	.27 .0417 .28	41	2A	02		AJMP	PCOF	; output char and return
	.27 .0417 .28	41	2A	02		AJMP	PCOF	; output char and return
	.27 .0417 .28	41	2A	02		AJMP	PCOF	; output char and return
	.27 .0417 .28	41	2A	02		AJMP	PCOF	; output char and return
	.27 .0417 .28	41	2A	02		AJMP	PCOF	; output char and return
	.27 .0417 .28	41	2A	02		AJMP	PCOF	; output char and return
	.27 .0417 .28	41	2A	02		AJMP	PCOF	; output char and return
	.27 .0417 .28	41	2A	02		AJMP	PCOF	; output char and return
	.27 .0417 .28	41	2A	02		AJMP	PCOF	; output char and return
	.27 .0417 .28	41	2A	02		AJMP	PCOF	; output char and return
	27 0417 28 29 0419	21	ÇA		KPC1:	AJMP AJMP	PCIF	; output char and return ; ELSE do local and return
	27 0417 28 29 0419	21	CA.		KPC1:	AJMP AJMP	PCIF	; output char and return ; ELSE do local and return
	27 0417 28 29 0419	21	CA.		KPC1:	AJMP AJMP	PCIF	; output char and return ; ELSE do local and return
	27 0417 28 29 0419	21	CA CA		KPC1:	AJMP	PCIF	; output char and return ; ELSE do local and return
	27 0417 28 29 0419	21	CA CA		KPC1:	AJMP	PCIF	; output char and return ; ELSE do local and return

			, ,	;	KBTAB -	keyboard tab routine	
. ,					*KBTAB*	checks to see if the S	SHIFT key is down. If it is ANSI mode. If it is in one
			••••••••••••		down th	en it checks for ZDS or	ANSI mode. If it is in one
					of thes	e modes then the sequer	nce to for backtab is sent.
			;				
			5		ENTRY	none	
					EXIT	none	
					USES	all	
			;		USES	e. 1 1	
					15.155	CHIEFE NT1	; IF not SHIFT THEN Just TAB
.041B	30 90	1C 0431	.ofK	BTAB:	JNB MÖV	SHIFTF, KT1 DPTR, #KTM1	; get ANSI sequence
041E 0421	22	40	06		JB	ANSI, KTO	; IF ANSI mode THEN do it
.9721	#X 30	41	. <u>06</u>		JNB	ZDS, KT1	; IF not ZDS THEN just TAB
0427	90	0434			MOV	DPTR, #KTM2	; ELSE set ZDS sequence
042A	02	OE6D	K	To:	LJMP	PSOF	; output back tab sequence
042D	74	 09		T1:	MOV	A, #HT	
042F	41	2A			AJMP	PCOF	; ELSE output horizontal tab
0401	1.B	5B	DA K	TM1:	DB DB	ESC, /[/, (/Z/+80H)	; ESC [Z ; ESC -
					<u>-</u> . <u>3</u>		
0431 0434	1B	AD	, K	TM2:		ESC, (2-2+80H) Ekeyboard.space.rout	
0434			, , , , , , , , , , , , , , , , , , ,	;	KBSPAÇI *KBSPA	Ekeyboard.space.rout	ine ess the control key is down
0434			, , , , , , , , , , , , , , , , , , ,		KBSPAÇI *KBSPA	Ekeyboard.space.rout	ine ess the control key is down
0434			, , , , , , , , , , , , , , , , , , ,	;	KBSPAC *KBSPA in whi	Ekeyboard.space.rout CE*.outputs.a.space.unl ch case it outputs a nu	ine ess the control key is down
0434			, , , , , , , , , , , , , , , , , , ,	.TM2:	KBSPAC *KBSPA in whi ENTRY	E — keyboard space rout CE* outputs a space unl ch case it outputs a nu none	ine ess the control key is down
0434			, , , , , , , , , , , , , , , , , , ,	;	KBSPAC *KBSPA in whi ENTRY	Ekeyboard.space.rout CE*.outputs.a.space.unl ch case it outputs a nu	ine ess the control key is down
0434			, , , , , , , , , , , , , , , , , , ,	;	KBSPAC *KBSPA in whi ENTRY	E.— keyboard space rout CE* outputs a space unl ch case it outputs a nu none	ine ess the control key is down
0434			, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,	;	KBSPAC #KBSPA in whi ENTRY EXIT	E.— keyboard space rout CE* outputs a space unl ch case it outputs a nu none	ine ess the control key is down
0434	18	AD	5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5	1	KBSPAC #KBSPA in whi ENTRY EXIT	E.—.keyboard.space.rout CE* outputs a space.unl ch case it outputs a nu .none .none all	ing ess the control key is down character.
0434	1B	AD	5 5 7 7 7 7 7	;	KBSPAC *KBSPA in whi ENTRY EXIT USES	E — keyboard space rout CE* outputs a space unl ch case it outputs a nu none none	ine ess the control key is down character. ; Default to space ; IF control space THEN
0434	1B	AD	;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	i	KBSPAC *KBSPA in whi ENTRY EXIT USES MOV JNB CLR	E — keyboard space rout CE* outputs a space unl ch case it outputs a nu none all A, #' / CONTLE, KBS1	ine ess the control key is down character. ; Default to space ; IF control space THEN ; output a null
0434	18 74 30 E4	20 1P	;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	1	KBSPAC #KBSPA in whi ENTRY EXIT USES MOV UNB	E — keyboard space rout CE* outputs a space unl ch case it outputs a nu none all A, #' / CONTLE, KBS1	ine ess the control key is down character. ; Default to space ; IF control space THEN
0434 	18 74 30 E4	20 1P	;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	i	KBSPAC *KBSPA in whi ENTRY EXIT USES MOV JNB CLR	E — keyboard space rout CE* outputs a space unl ch case it outputs a nu none all A, #' / CONTLE, KBS1	ine ess the control key is down character. ; Default to space ; IF control space THEN ; output a null
0434 	18 74 30 E4	20 1P 2A	, , , , , , , , , , , , , , , , , , ,	; (BSPACE: (BS1:	KBSPAC *KBSPA in whi ENTRY EXIT USES MOV UNB CLR AJMP	E keyboard space rout CE* outputs a space unl ch case it outputs a nu none all A, #' CONTLE, KBS1 A PCOF	ine ess the control key is down ll character. ; Default to space ; IF control space THEN ; output a null ; Output character and return
0434 	18 74 30 E4	20 1P 2A	, , , , , , , , , , , , , , , , , , ,	; (BSPACE: (BS1:	KBSPAC *KBSPA in whi ENTRY EXIT USES MOV UNB CLR AJMP	E keyboard space rout CE* outputs a space unl ch case it outputs a nu none all A, #' CONTLE, KBS1 A PCOF	ine ess the control key is down character. ; Default to space ; IF control space THEN ; output a null
0434 	74 39 E4	20 1B 2A	5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,	; :BSPACE: (BS1:	KBSPAC *KBSPA in whi ENTRY EXIT USES MOV JUNE CLR AJMP	E — keyboard space rout CE* outputs a space unl ch case it outputs a nu none all A, #' CONTLF, KBS1 A PCOF	ine ess the control key is down character. ; Default to space ; IF control space THEN ; output a null ; Output character and return
0434 	74 30 E4 41	20 1P 2A	5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5	: (BSPACE: (BS1:	KBSPAC #KBSPA in whi ENTRY EXIT USES MOV JNB CLR AJMP	E keyboard space rout CE* outputs a space unl ch case it outputs a nu none all A, #' CONTLF, KBS1 A PCOF	ine ess the control key is down ll character. ; Default to space ; IF control space THEN ; output a null ; Output character and return

				RMWARE, VEI			
• • • • • • • • • • • • • • • • • • • •	··· <u>1</u>	• • • • • • • • • • • • • • • • • • • •				KBENT	- keyboard enter routine
	3					*KBENT	* outputs a carriage return unless the shift key is
	4 5				5	down i	n which case it does the print page function.
*********	6		***********	• • • • • • • • • • • • • • • • • • • •	· · · · · · · · · · · · · · · · · · ·	• • • • • • • • • • • • • • • • • • • •	
	7					ENTRY	none
	9 				7	EXIT	Done
					***************************************		none
• • • • • • • • • • • • • • • • • • • •	<u>11</u>					USES	all
	13						
	14 043E 15 0441	20 7 4	1D 9B	04	KBENT:	JB.	CONTLF, KBE1 ; IF no control THEN
• • • • • • • • • • • • • • • • • • • •	16 0443	···· <u>/</u> 47·····	öF			YOM AJMP	A, #KB_ENT ; get kepad enter key KP2 : enter back into code
	17 0445 18	02	1030		KBE1:	JMP	KP2 ; enter back into code R2_PRNT ; ELSE do print page function
	18 19						
	19 20		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	
	<u>21</u>					· · · · · · · · · · · · · · · · · · ·	
	. 23 				; ;	KBACC -	- keyboard access routine
	24				,	*KBACC	* is run whenever the access character is received
• • • • • • • • • • • • • • • • • • • •	. 25 . 26	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		from th	he keyboard.
	27 				, ,		
	28 29				;	ENTRY	none
	. <u>29</u> . 30			• • • • • • • • • • • • • • • • • • • •	· · · · · · · · · · · · · · · · · · ·	EXIT	none
• • • • • • • • • • • • • • • • • • • •	.31 .32					· · · · · · · · · · · · · · · · · · ·	
	33				,	USES	none
	34 35 0448	D2	1E		terace.		
• • • • • • • • • • • • • • • • • • • •	36 044A	22	15	• • • • • • • • • • • • • • • • • • • •	KBACC:	SETB RET	ACCESSF ; Flag access received
						1100	
• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •		

•••••	• • • • • • • • • • • • • • • • • • • •						
			••••••				
						• • • • • • • • • • • • • • • • • • • •	
		• • • • • • • • • • • • • • • • • • • •					

		MWARE YERSION 1.03.**		- keyboard data tabl	e
. 1 2		 ; ;	• • • • • • • • • • • • • • • • • • • •		nds received from the keyboard
.3 4					ard data word followed by the
.5 6			addre:	entries are:Nexuy ss of the routine wh	ich performs the requested function.
 8 9	044B	KBDT	EQU	\$	
10			DB	KB_TAB	; Tab key
11 044B 12 044C	8B 041B		ĎŴ	КВТАВ	
13	 8C		DB	KB_SPACE	; Space bar
14 044E 15 044F	0436		DW	KBSPACE	
16	OF.		DВ	KB_ACC	; Access code
179451 18 0452	9F 0448		ĎW	KBACC	
19 20 0454	 8F		DB	KB_POWER	; Keyboard just powered up
20 0454 21.0455	101E		₽₩	R2_IKB	
22 23.0457	89			KB_BREAK	; Break pressed
24 0458	OFCC		DW	KBBREAK	
25 26 045A	 8A		DB	KB_CAPS	; Caps locked mode
27. Q45B.	OFBF		DW	KBCPLK	
28 29. 945D.,	87		pB	KB_SCROLL	; Scroll key Pressed
30 0 45 E	OFE3		DW	KBSCROLL	
.31 32 0460	88		DB	KB_SETUP	; Setup key pressed
33.9461	1021		DW	R2_SETUP	••••••
34 35. 0463 .	9B		DB	KB_ENT KBENT	; Enter key pressed
36 046 4 37	043E		DW	KBENI	
.\$./ 38	0009	KBDTL	EQU	(\$-KBDT)/3	; Table length
				. ,,	

)

A. A. A. A. A. A. M.	WIER TERMINA	ROSS ASSEMBLER (V1)-218 L.FIRMWARE VERSION 1.03	24-FEB-83 1 .***	5:30:06 PAGE 40	
1					
2		* 5	KBCRT	- keyboard cursor	table
4		· · · · · · · · · · · · · · · · · · ·	*KBCR	T* contains the ASC	II values to be transmitted by the
5			termi	nal.when.received.fi	rom the keyboard.
ž		,	Table	entries are: Keyb	pard data word followed by the
 		7	ASCII	character to be tra	ansmitted if in shift made followed by
10 11		**************************************	here	is the erase and sh	transmitted always. The special case ift erase function key.
12 13	0466	KBCRT	EQU	\$	
14 15.0466	90			***************************************	
16 0467	80 00 4:	1	DB DB	KB_UP 0, 'A'	; Up arrow key
<u>17</u> 18 0469	 81		DB		
	9042	2	DB DB	KB_DOWN O, /B/	; Down arrow key
20					
21.046C 22.046D	\$2 00 44		DB DB	KB_LEFT	; Left arrow key
23					
24 046F 25 0470	83 00 43	3	DB DB	KB_RIGHT	; Risht arrow key
26			µ₽	99	
279472 28.0473	84	······	DB DB	KB_HOME	; Home key
29 30 0475				0, 7H7	
30 0475 31 0476 .	86 00 7E	=	DB	KBLHELP	; Help key
32		-	bB	0, /~/	
33.0478 34.0479	85 45 4 <i>6</i>	······	DB DB	KB_ERASE	; Erase key
35 36				'E', 'J'	
36	0007	KBCRTL	EQU	(\$-KBCRT)/3	; Table length
		•••••	• • • • • • • • • • • • • • • • • • • •		•••••••••••••••••••••••••••••••••••••••
	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		
		• • • • • • • • • • • • • • • • • • • •			
				• • • • • • • • • • • • • • • • • • • •	
• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			·····
• • • • • • • • • • • • • • • • • • • •		•••••	• • • • • • • • • • • • • • • • • • • •		
				•••••	
		• • • • • • • • • • • • • • • • • • • •			
• • • • • • • • • • • • • • • • • • • •					

047B 3 047B 90 9 047C 30 70 047E 91 2 047F 31 71 3 0481 92	; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	*KBPDI termin Table ASCII by the The si follou transm	entries are: Key character to be t e ASCII character hifted key format wed by teh escape mitted. Thee the	CII values to be transmitted by the from the keyboard. board data word followed by the ransmitted if in unshifted followed to be transmitted if in any mode. uses keyboard data + shift function sequence or ASCII character to be tion to this is the insert character for the second is for leaving IC mode.
047B 047B 90 047C 30 70 047C 31 71 047F 91 71	; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Table ASCII by the The s! follow trans: key. enter: EQU DB	entries are: Key character to be t e ASCII character hifted key format wed by teh escape mitted. The excep In this case the ing IC mode and th	board data word followed by the ransmitted if in unshifted followed to be transmitted if in any mode. uses keyboard data + shift function sequence or ASCII character to be tion to this is the insert character first character is the character for
047B 047B 90 047C 30 70 047C 31 71 047F 91 71	; ; ; ; ; ; ; ; ; ; KBPDT	ASCII by the The s! follow transmer key. enter EQU DB	character to be te ASCII character hifted key format wed by teh escape mitted. The exception this case the ing IC mode and the	ransmitted if in unshifted followed to be transmitted if in any mode. uses keyboard data + shift function sequence or ASCII character to be tion to this is the insert character first character is the character for
047B 047B 90 047C 30 70 047C 31 71 047F 91 71	; ; ; ; ; ; ; ; KBPDT	ASCII by the The s! follow transmer key. enter EQU DB	character to be te ASCII character hifted key format wed by teh escape mitted. The exception this case the ing IC mode and the	ransmitted if in unshifted followed to be transmitted if in any mode. uses keyboard data + shift function sequence or ASCII character to be tion to this is the insert character first character is the character for
047B 3 047B 90 9 047C 30 70 047C 91 047F 91 71	; ; ; ; ; KBPDT	The st follow transmers key- entermers EQU	hifted key format wed by teh escape mitted. The except In this case the ins IC mode and the	uses keyboard data + Shitt function sequence or ASCII character to be etion to this is the insert character first character is the character for
047B 3 047B 90 9 047C 30 70 047C 91 047F 91 71	; ; ; ; KBPDT	follou trans: kev. enter: EQU	wed by teh escape mitted. The excep In this case the ing IC mode and th	sequence or ASCII character to be tion to this is the insert character first character is the character for
047B 047B 90 047C 30 70 047E 91 047F 31 71	; ; ; KBPDT	ker. enter: EQU DB	In this case the ins IC mode and th	first character is the character for
047B 047B 90 047C 30 70 047C 31 71	, KBPDT	enter: EQU DB	ing IC mode and th	e second is for leaving IC mode.
047B 047B 90 047C 30 70 047C 91 047F 91 047F 31 71	КВРДТ		· · · · · · · · · · · · · · · · · · ·	
047B 90 047C 30 79 047C 91 71 047F 91 71	КВРОТ		· · · · · · · · · · · · · · · · · · ·	
047B 90 047C 30 70 047E 91 047F 31 71			КВ_0	
047C 30 70 70 70 94 70 94 70 94 70 94 71 71 71 71 71 71 71 71 71 71 71 71 71		no		; Keypad O key
047E 91 2 047F 31 71				
2 047F 31 71 3		DB	KB_1	; Keypad 1 key
		DB	/1/, /q/	
		DB	KB_2	; Keypad 2 key
5.048232		p.B	(2(x/t/	
•		DB	кв_3	; Keypad 3 key
7.048493 3.0485 33 73	• • • • • • • • • • • • • • • • • • • •	DB	KB_3 /3/, /s/	
9 0.0487 94		DB	КВ_4	; Keypad 4 key
) 0487 94 L 04883474		DB		
2		DB	KB_5	; Keypad 5 key
3.Q48A25 4 048B 35 75		DB	KB ₊ 5 /5/, /u/	
5		DB	кв_6	; Keypad 6 key
5 048D 96 7 . 048E 3676		DB		
3		DB	KB 7	s Keypad 7 key
9. Q49097 0. Q491 — 37 — 77		DB	KB ₊ 7 ′7′, ′w′	
1				; Keypad 8 key
2 0493 98 3 <u>0494</u> 38		DB	1813X	
4		n.r.	עם ס	; Keypad 9 key
5. 0496		DB	KB_9	
Z				; Keypad decimal point key
8 0 4 99 9A		DB DB	KB_DEC 1.1.101	> Keirad decimal roune ser
9049A2E6E 0				* Kaupad anton kay
1.049C2B		DB DB	KB_ENT CR, /M/	; Keypad enter key
2 049D OD 4D 3				
4 049F 9C 5 04AQ 2D 6D		DB DB	KB_DASH	; Kerpad dash (minus) ker

	1.04A3	2C	ĄÇ		DB	4,4, 4)4	
	2 3.94A5	DΩ					••••••
	9.9769 4.04A6	Þo 30	00		DB DB	KB_0 + KB_SHFT	; Keypad O key shifted
	5				DB	O', NULL	
	6 04A8	D1			DB	KB_1 + KB_SHFT	; Keypad 1 key shifted
	79402 8	1B	4Ų	• • • • • • • • • • • • • • • • • • • •	DB	ESC, /L/	
	2.04AB				DB	KB_2 + KB_SHFT	• Karra I O I
	0 04AC	1 B	42		DB	ESC, 'B'	; Keypad 2 key shifted
	1 2 04AE	D3	• • • • • • • • • • • • • • • • • • • •		<u></u>	***** <u>**</u> ******************************	
	3.94AF		4D		DB DB	KB_3 + KB_SHFT ESC, 'M'	; Keypad 3 key shifted
1	4			••••••••••••		5991 IT	
	504B1	D4 1B			DB	KB_4 + KB_SHFT	
	6 04B2 7	18	44		DB	ESC, 'D'	
	8 04B4	D5	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	DB	KB_5 + KB_SHFT	Waller F. W.
	9.94B5	1.B	48		DB	ESC, 'H'	; Kerpad 5 ker shifted
2	O 1_04B7	<u>p</u> 6					
	2 04B8	1B	43	• • • • • • • • • • • • • • • • • • • •	DB	KB-6.±.KB-SHFT ESC, 1C1	
	3		• • • • • • • • • • • • • • • • • • • •		מט	E3U, 1U	
	4 04BA 5 04BB	D7			DB	KB_7 + KB_SHFT	; Keypad 7 key shifted
20		1.B	49		DB	ESC, /@/	***************************************
	7. 04BD	ps			DB	KB_8 + KB_SHFT	* Maxima d. C. Joseph School
	3 O4BE	1B	41		DB	ESC, A	; Keypad 8 key shifted
·····.43	9 0 04C0	 D9	• • • • • • • • • • • • • • • • • • • •			· · · · · · · · · · · · · · · · · · ·	
3		1B	4E		DB DB	KB_9 + KB_SHFT ESC, /N/	; Kerpad 9 ker shifted
31		*** A					•••••
	3.04C3 3.04C4	PA 2E	 00		DB	KB_DEC + KB_SHFT	; Keypad decimal key shifted
35	5				DB	A. A. NULL	
	0406	DB		***************************************	DB	KB_ENT + KB_SHFT	; Keypad enter key shifted
	7. 94C7	<u>od</u>	00		DB	CR, NULL	, well do enter ker shifted
	0409	DC			DB	KD DACH - KD OUET	
40	0 04CA	2D	00	***************************************	DB	KB_DASH + KB_SHFT	; Keypad dash key shifted
4]		DD			DB		
43	04CD		00			KB_COMMA + KB_SHFT	; Kerpad comma ker shifted
44	ļ				pB		
45	j	001C		KBPDTL	EQU	(\$-KBPDT)/3	; Table length
			• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		
			• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		
	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • •	•••••••	••••••	• • • • • • • • • • • • • • • • • • • •		

C C

2 3 4 5 6 7 8 9 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0	53	; ; ; ; ; ; KBFNT	*KBFNT termir Table	nal when received from entries are: Keyboar character to be trans	values to be transmitted by the
4 5 6 7 8 9 0 1 0 4 0 4 D 0 5 3 1 5 6 0 4 D 2 8 1 1 7 0 4 D 3 5 4 1 8 9 0 4 D 5 8 2 8 9 9 0 4 D 5 8 2 9 0 4 D 5 8 2 9 9 0 4 D 5 8 2 9 0 0 4 D 5 8 2 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	53	KBFNT	termir Table ASCII by the EQU DB	entries are: Keyboar character to be trans ASCII character to be	the keyboard. d data word followed by the mitted if in ANSI mode followed e transmitted in any other mode.
0 04 04 04 04 04 04 04 04 04 04 04 04 05 05 05 05 05 05 05 05 05 05 05 05 05	53	; ; ; KBFNT	ASCII by the EQU DB	character to be trans ASCII character to be	mitted if in ANSI mode followed e transmitted in any other mode.
0 04 04 04 04 04 04 04 04 04 04 04 04 05 05 05 05 05 05 05 05 05 05 05 05 05	53	, , KBFNT	BY the	* ASCII character to b	e transmitted in any other mode.
0 04 04 04 04 04 04 04 04 04 04 04 04 05 05 05 05 05 05 05 05 05 05 05 05 05	53	KBFNT	EQU		
1 04 2 3 04CF 80 4 04D0 53 15 16 04D2 81 17 04D3 54 18	53	KBFNT	DB		: Function key #1
2 3 04CF 80 4 04D0 53 16 04D2 81 17 04D3 54 18	53		DB na		: Function key #1
4 04D0 53 5 6 04D2 81 17 04D3 54 18 19 04D5 82	53		ñā		
16 04D2 81 17 04D3 54 18 19 04D5 82				787, 787	; ESC O S, ESC S
17 04D3 54 18 19 04D5 82			pB	KB_F2	; Function key #2
9 04D5 82			DB	΄Τ΄,, ΄Τ΄	; ESC O T, ESC T
			DB	KB_F3 ∵′′′′′′′′′′′′′′′′′′′′′′′′′′′′′′′′′′′′	; Function key #3 ; ESC O U, ESC U
20 04D6 55 21	55		DB	707, 707	
22 04D8 83			DB DB	KB_F4 /V/, /V/	; Function key #4 ; ESC O V, ESC V
23 0 4D 9 56 2 4					
25 04DB 84 26 04DC 57	5 7		DB DB	KB_F5 /W/, /W/	; Function key #5 ; ESC O W, ESC W
27			DB	KB_F6	; Function key #6 (blue)
28 04DE 85 29 04DF 50			DB	/P/, /P/	; ESC O P, ESC P
30			DB	KB_F7	; Function key #7 (red)
31 04E1 86 32 04E2 51	?5i		DB	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	; ESC O Q, ESC Q
33 34 04E4 87	,		ĎB	KB_F8	; Function key #8 (gray)
35 04E5 52			DB	′R′, ′R′	; ESC O R, ESC R
36 37 04E7 - 88 38 04E8 - 58	3		DB	KB_F9	; Function key #9 ; ESC O X, ESC O I
	c9	,	DB	/X/, (/I/+80H)	
39 40 00	009	KBFNTL	EQU	(\$-KBFNT)/3	; Table length

					::	DI TNIAT	9 - Kuzga 92		
2			• • • • • • • • • • • • • • •		***************************************	SCTWHI	Y.T. Dullaineadanes	ş	•••••
3	• • • • • • • • • • • • • • • • • • • •					*BLINA	AD* calculates charac	ter memory addresses using _vpoc_	
5					i i	4114	n oo and biaces (Ne	result into -LINADH/L	
6					***************************************				
	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			EŅŢŖY.	none		
<u>.</u>					; ;				
10					***************************************				• • • • • • • • • • • • • • • • • • • •
12	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		UŞEŞ	A, PTR1		
13									
	04EA 04EC	CO	FO		BLINAD:	PUSH	В	; Save register	
	04EE	74 25	47	• • • • • • • • • • • • • • • • • • • •		Mov	A. #LORDER		
	04F0	F8	12			ADD	ni irua	; Add in offset	
18	04F1	E6	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	MOV MOV	PTR1, A A, @PTR1		
	04F2	75	FO	50		MOV	B, #NUMCHARS	; retch pointer	
	04F5	A4		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	MUL	AB		
	.04F6	25	11			ADD		· Add in Mood of .	
	04F8	F5	14			MOV	LINADL, A	; Save low order byte	
	.04FA	Ę5 34	F0 50		• • • • • • • • • • • • • • • • • • • •	MQV	A B		
	04FE	F5	13			ADDC	TO THE TOTAL COMMISSION	, Huu Carry and offset	
	0500	DO	‡¤ F0	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	MOV	LINADH, A		
						POP	В	; Restore	
	0502	22				RET			
	0502	22		•••••••••••••••••••••••••••••••••••••••					•••••
	0502	22							
		22							
	0502	22							
	0502	22							
	0502	22							
	0502	22							
	0502	22							
	0502	22							
	0502	22							
	0502	22							
	0502								
	0502	22							
	0502	22							

	729ROM; MICROB *** Z-29 COMPU	TER TERMINAL	FIRMWARE VE	RSION 1.03 ***			
S	1					calculate line address	
5 ENTRY (A) = X-mosition 5 EXIT (DPTR) = current address 11 USES A. B. DPTR, TEMP. PTRI 12 USES A. B. DPTR, TEMP. PTRI 13 OSO3, FB. CLA! MOV TEMP. A. I. Save -XFOS 14 OSO3, FB. CLA! MOV TEMP. A. I. Save -XFOS 15 OSO3 FB. C. CLA! MOV TEMP. A. I. Save -XFOS 16 OSO3 FB. C. CLA! MOV TEMP. A. I. Save -XFOS 17 OSO5 FS. FO. ADD. A. B. I. Add.inYFOS -OSTSE! 18 OSO5 FS. FO. ADD. A. B. I. Add.inYFOS -OSTSE! 19 OSO5 FS. FO. ADD. A. B. I. Add.inYFOS -OSTSE! 19 OSO5 FS. FO. ADD. A. B. I. Add.inYFOS -OSTSE! 19 OSO5 FS. FO. ADD. A. TEMP Add in -XFOS -OSTSE! 21 OSO5 ED. A. B. I. ADD. A. TEMP Add in -XFOS - OSTSE! 22 OSO5 ED. A. ADD. A. TEMP Add in -XFOS - OSTSE! 23 OSOF FS. S. S. MOV - MPL. A. B. I. D., I. Add. IN-POS - OSTSE! 24 OSI1 ES FO. ADD. A. HAION.CHARMEN I. Add. IN-POS - OSTSE! 25 OSI3 ED. ADD. A. TEMP Add in -XFOS - OSTSE! 26 OSI3 ED. ADD. A. TEMP ADD. A. TEMP ADD. A. TEMP 27 OSI3 ED. ADD. A. TEMP ADD. A. TEMP ADD. A. TEMP 28 OSI3 ED. ADD. A. TEMP ADD. A. TEMP ADD. A. TEMP 29 OSI3 ED. ADD. A. TEMP ADD. A. TEMP 20 OSI3 ED. ADD. A. TEMP ADD. A. TEMP 21 OSO3 ED. ADD. A. TEMP ADD. A. TEMP 22 OSO5 ED. A. A. TEMP ADD. A. TEMP 23 OSO5 ED. A. A. TEMP ADD. A. TEMP 24 OSI3 ED. A. TEMP ADD. A. TEMP 25 OSI3 ED. ADD. A. TEMP ADD. A. TEMP 26 OSI3 ED. A. TEMP ADD. A. TEMP 27 OSI3 ED. A. TEMP ADD. A. TEMP 28 OSI3 ED. A. TEMP ADD. A. TEMP 29 OSI3 ED. A. TEMP ADD. A. TEMP 20 OSI3 ED. A. TEMP ADD. A. TEMP 21 OSO5 ED. A. TEMP ADD. A. TEMP 22 OSO5 ED. A. TEMP ADD. A. TEMP 23 OSO5 ED. A. TEMP ADD. A. TEMP 24 OSO5 ED. A. TEMP ADD. A. TEMP 25 OSO5 ED. A. TEMP 26 OSO5 ED. A. TEMP 27 OSO5 ED. A. TEMP 28 OSO5 ED. A. TEMP 29 OSO5 ED. A. TEMP 20 OSO5 ED. A. TEMP 21 OSO5 ED. A. TEMP 22 OSO5 ED. A. TEMP 23 OSO5 ED. A. TEMP 24 OSO5 ED	2			;	*CLA*	inputs line coordinates	and calculates the current
State Stat	4 5			,	addres	s of that coordinate to	character memory.
B	·····				ENTEV	$(\Delta) = Y - position$	
11 12 15 15 15 15 15 15	7 8					(B) = Y-position	
12	9				EXIT	(DPTR) = current addr	ess
14	12				USES	A, B, DPTR, TEMP, PTR	i
1. 0804							. C VPOC-
17,0506		FD		CLA:		A. #LORDER	; Get base of table
18 0508 F8 MOV PIRL H I NOV PIRL H I NOV FIRL H I NOV FIR					ADD	A, B	; Add in -YPOS- offset
7.0 7.0	18 0508	F8				PTR1, A	
20.0500 A4 MUL AB I.Do. it 1 Add in -XPOS- offset 20.050		<u>E6</u>					; Set up for multiply
22 OSOF F5 82 MQV DPL.A 1.Save.low.bxte. 22 OS1 E5 F0 MQV A B 1 Get high byte 22 OS1 F5 83 MQV DPH.A 1.Save.low.rx.effset.and.carry. 22 OS1 F5 83 MQV DPH.A 1.Save.high byte 27 OS17 F2 83 MQV DPH.A 1.Save.high byte 27 OS17 22 SET	20 050A		-				; Do it
23 050F F5 82 MOV ALB 1 5840 Section byte 25 0513 34 50 ADDC ALBERT Section byte 26 0515 F5 83 MOV DPH A Save high byte 27 0517 22 RET	22 050E	2D			ADD	A, TEMP	; Add in -XPOS- offset
24 0515 E5 F0 ABDC A, HIGH (CHARMEN) I Add memory effect and carry 26 0515 F5 83 MOV DPH, A I Save high byte 87 0517 22 REI		F584					5 Save low DYTS
26 0315 F5 63 MOV DPH, A ; Save high byte 27 0317 22 RET	24 0511						; Add memory offset and carry
27.0817 22 RET.							
			5				

2	• • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		PRLF.	m. Perform a reverse 1	ine feed
<u>.</u>			; ;			.one.line in the video memory.
4			\$	TT CH	e cursor was aiready	ON the top line of the display.
6	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		ፕ ቦድ	lsPlax.ls.scrolled.do	wn and the new line is filled
7			,	witn	spaces.	
				ENTRY	none	
				EXIT	none	
13			;	Horo		
14		• • • • • • • • • • • • • • • • • • • •		USES	all	
15 16 0518	<u></u>					
16 0518 17 051A		12	PRLF:	MÓV	A, YPOS	; Get Y-position
18 051	B5 85	17 18		CJNE	A, TSCROLL, RL1	; IF top of scroll region THEN
19 0520		18 F0	FO	MOV	B, BFIX	; clear bottom line
20 0522		F9 0B96	• • • • • • • • • • • • • • • • • • • •	DEC	B	of the scrolling region
21 0525				LCALL	CLRLINE	
22 0527	25	47 18	••••••	MOV ADD	A, #LORDER	; Get top of table
23 0529	14			ADD DEC	A, BFIX A	; Add in offset
24 052A	F8		• • • • • • • • • • • • • • • • • • • •	MÖV	PTR1, A	
25 052B	F9			MOV	PTR2, A	
26 0520	19		• • • • • • • • • • • • • • • • • • • •	DEC	PTR2	
27_052D	86 E5	05 18		MOV	TEMPA, @PTR1	. Com ball of the second
28 052F				MÖV	A, BFIX	; Save bottom to put on top
29 0531	95	17		SUBB		
30 0533	14			DEC	A, TSCROLL A	; Calc # times through loop
31 0534	FC			MOV	WORK, A	, caic # times through loop
32 0535	<u>C2</u>	AF		CLR	EA	; Disable for flicker
33 0537 34 0538	E7 F6		RLL:	MOV	A, @PTR2	; REPEAT
34 0338 35 0539				MOV	ertri, A	; exchange values
35 0537 36 053A	18 19		• • • • • • • • • • • • • • • • • • • •	DEC	PTR1	
37 053B	DC	FA		DEC	PTR2	; bump down
38				DJNZ	WORK, RLL	; UNTIL work = 0
39 053D	ΔA	05				
40 053F	A6	AF		MOV	@PTR1, TEMPA	Fut last value in
41 0541	E5			SETB	ÉA	; Enable interrupts again
42 0543	·····řš···	$\dots \frac{12}{12}\dots$	RLR:	MOV MOV	A, YPOS	; Keep same position
43 0545	91	ĒĀ	_\\.		YPOS, A	
44 0547	22		• • • • • • • • • • • • • • • • • • • •		DETIME	5 Build line address
45 0548	B4	00	01 RL1:		Δ. #O. RI1Δ	
			•••••		······································	3 IF TOP OF SCREEN THEN
47 054C	14		RL1A:		A	
48 0540	80	F4		······ŠJMp···		
44 0547 45 0548 46 0548	22 B4 22			ACALL RÉT CJNE RÉT DEC SJMP	BLINAD A, #0, RL1A A RLR	; Build line address ; Return ; IF top of screen THEN ; return ; ELSE bump up ; and return

(

1						PCRLF -	- perform carriage	return and/or line feed
2					; ;	*PCRLF*	performs a carria	ise return and then performs ine feed function is selected.
4 5					,	a line	feed if the auto l	ine feed function is selected.
6 7					,	ENTRY	none	
						EXIT	none	
10					; 	USES	A, B, DPTR, TEMP,	PTR1
12 13.						ACALL		; Perform carriage return
15.	054F 0551	B1 20	5C 2D	<u>o</u> p	PCRLF:	JB RET	AUTOLE: PLF	; IF auto LF THEN do *PLF* ; ELSE return
17.	0554	22					,	, Else return
18 19.								
20 21.						P.L.F.CR . .	n.menform.line.feed	d_and/or_carriage_neture
22 23.						*PLFCR	* performs.a.carria	age return prior to performing a
24 25.							sed it the anto car	
26 27						ENTRY	none	
28 29.						EXIT	none	
30 31.						V\$E\$	A»B«PRTR«TEMP»	. PTR1
32 33.	 0555	30	2C	09	PLFCR:	JNB	AUTOCR, PLF	; IF no auto CR THEN do *PLF*
35.		B1 80					PCR	; ELSE do carriage return ; line feed and return
				· · · · · · · · · · · · · · · · · · ·				

)

)

Z29R ***	OM; MICRO Z-29 COMPL	BENCH 805 UTER TERM	1 CROSS UNAL FIR	ASSEMBLE MWARE VE	R (V1)-2 RSION 1.	218 24-F Q3.***	EB-83 15	5:30:06 PAG	iE 48						. 	•••••
	1	· · · · · · · · · · · · · · · · · · ·					PCR -	.perform.ca	rriage ret	urn						
	3				; 				cursor to		ning of t	he curre	nt line.			
					.											
	6 7				; ;		ENTRY	none						• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • •	
,	8 ?				;		EXIT	none	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •			•••••	
	10 11 12				;		USES	Α	**************			• • • • • • • • • • • • • • • • • • • •				
	12 13 055C 14 055F	75	11	00	PCR:	• • • • • • • • • • • • • • • • • • • •	MOV	YP09. #0	••••••							• • • • • • • • •
	14 055F	75 81	11 EA			• • • • • • • • • • • • • • • • • • • •	MOV AJMP	XPOS, #0 BLINAD	• • • • • • • • • • • • • • • • • • • •	······;	eturn to uild line	address	and ret	urn		•••••
		• • • • • • • • • • • • • • • • • • • •						• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	
	• • • • • • • • • • • • • • • • • • • •		•••••			• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •							• • • • • • • • • • • • • • • • • • • •
************			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •				
••••••			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			••••••		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	
		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	••••••••				• • • • • • • • • • • • • • • • • • • •			
••••••		•••••	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		•••••	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	
•••••	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	•••••••		••••••					•••••		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
•••••		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •				• • • • • • • • • • • • • • • • • • • •	
		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •							
• • • • • • • • • • • • • • • • • • • •	•••••	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •									
• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •		•••••		•••••		• • • • • • • • • • • • • • • • • • • •						
• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •													
••••••••	• • • • • • • • • • • • • • • • • • • •								••••••							
	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •				• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •						
• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •															
•••••	•••••															
	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •														
											• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •															
															• • • • • • • • • • • • • • • • • • • •	
• • • • • • • • • • • • • • • • • • • •						• • • • • • • • • • • • • • • • • • • •					• • • • • • • • • • • • • • • • • • • •					
					• • • • • • • • • • • • • • • • • • • •						• • • • • • • • • • • • • • • • • • • •					

C

### ### ##############################	
fixed region. The cursor never moves into the 25th line.	
ENTRY	
EXIT none	
EXIT none	
Section Sect	
Solicity	
12	
7 0563 B4 18 01 CJNE A, #MAXLINE, PLF1 ; IF on 25th line THEN 8 0566 22 RET ; return 9 0567 04 PLF1: INC A ; ELSE bump value 0 0568 B5 18 02 CJNE A, BFIX, PLF2 ; IF bottom scroll region Th 1 056B 80 04 SJMP PLF3 ; scroll as usual 2 056D B4 18 24 PLF2: CJNE A, #MAXLINE, PLFRET ; IF not 24th line THEN bump 3 0570 22 RET ; ELSE return 4 0571 85 17 F0 PLF3: MÖV B, TSCROLL ; Set up to clear 5 0574 12 0896 LCALL CLRLINE ; Top line of scroll region	
3 0566 22	
0567	
18 02 CJNE A, BFIX, PLF2 F bottom scroll region The control of the co	
L 056B 80 04 SJMP PLF3 ; scroll as usual 2 056D 84 18 24 PLF2: CJNE A, #MAXLINE, PLFRET ; IF not 24th line THEN bumb 3 0570 22 RET ; ELSE return 4 0571 85 17 FO PLF3: MÖV B, TSCROLL ; Set up to clear 5 0574 12 0896 LCALL CLRLINE ; Top line of scroll region	IEN
3 0570 22 RET ; ELSE return 3 0571 85 17 FO PLF3: MOV B, TSCROLL ; Set up to clear 5 0574 12 0896 LCALL CLRLINE ; Top line of scroll region	
1 0571 85 17 FO PLF3: MOV B, TSCROLL ; Set up to clear 5 0574 12 0896 LCALL CLRLINE ; Top line of scroll region	•
0.0574 12 0B96 LCALL CLRLINE ; Top line of scroll region	
TO THE TAX AND THE	
0579 25 17 ADD A, TSCROLL ; For adjusting -LORDER- 057B F8 MOV PTR1, A	
0576 F8 MOV PTR2, A	
057D 09 INC PTR2	
0580 E5 18 MOV A, BFIX	
0582 95 17 SUBB A, TSCROLL ; Calc number of times	
OS84 14 TEC A TO 90 Through 100P	
i 0585 FC MOV WORK, A Stops any blinking	
- 0000 C2 H	
OS88 E7 PLFLP: MOV A, @PTR2 ; REPEAT	
O589 F6 MOV @PTR1, A ; switch entries	
OSSA OS INC PTR1	
0588	
058E A6 05 MOV @PTR1, TEMPA ; Place last value	
· UJOE HO VO	
0592 F5 12 MOV A, YPOS ; Keep same position	
7 0594 F5 12 PLFRET: MOV YPOS, A ; Updata -YPOS- 3 0596 81 EA AJMP BLINAD ; Build line address and re	urn

Z29ROM; MICROBENCH 8051 CROSS ASSEMB *** Z-29 COMPUTER TERMINAL FIRMWARE.	_ER (V1)-218 24-FE VERSION 1.03 ***	B-83 15	:30:06 PAGE 50	
1		CLRTABS	Sclear all tabs	
2 4			38*.cleans.all.tab.setti	ings.
<u>\$</u>	,	ENTRY	none	
	· · · · · · 5 · · · · · · · · · · · · · · · · · · ·	EXIT	none	
9 10 11		USES	A, PTR1, INDX1	
12 13 0598 78 29 14 059A E4 15 059B 7A 0A	CLRTABS:	MOV CLR MOV	PTR1, #TABTAB A INDX1, #10	; Init pointer to tab table ; Init value to place in table ; Ten bytes in the tab table
17 059D F6 18 059E 08 19 059F DA FC 20 05A1 22	CLRTB:	MOV INC DUNZ RET	@PTR1, A PTR1 INDX1, CLRTB	; REPEAT zero byte ; bump pointer ; UNTIL index = O
	•••••	•••••		
	••••••	•••••		
			• • • • • • • • • • • • • • • • • • • •	
	•••••			
			••••••	
	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		
			•••••	
	• • • • • • • • • • • • • • • • • • • •			•••••••••••••••••••••••••••••••••••••••
	• • • • • • • • • • • • • • • • • • • •			
				·····
	• • • • • • • • • • • • • • • • • • • •			

Z29ROM; MICROBENCH 8051 CROSS ASSEMBL *** Z-29 COMPUTER TERMINAL FIRMWARE V	ER (V1)-218 24 ERSION 1.03 ***				
1		SETTAI	B — chanse tab settins		
2 3 4 5	; ; ;	*SETT/	AB* sets or clears a t s passed to it.	ab position depending on the	
	\$ \$ \$	ENTRY	(A) = tab position 'FO' = value to pla	from 079 ce into tab position	
		EXIT	none		
11 12 13	,	·····UŚEŚ	A, B, PTR1, WORK, I	NDX1	
14 15 05A2 F5 F0 16 05A4 F8 17 05A6 FC 18 05A7 03 19 05A8 03 20 05A9 03 21 05AA 24 29 22 05AC F8 23 05AD E6 24 05AE 7A 08 25 05B0 C5 F0 26 05B2 B 74 27 05B5 A2 D5 28 05B7 92 F7 29 05B9 C5 F0 30 05BB 23 31 05BC DC 32 05BD DA F1	SETTAB: STB1: STB2:	MOV ANL MOV RR RR RR ADB MOV MOV MOV XCH CUNE MOV XCH RL INC	B, A A, #11111000B WORK, A A A A A A, #TABTAB PTR1, A A, @PTR1 INDX1, #8 A, B A, WORKA, STB2 C, F0 B.7, C A, B A WORK INDX1, STB1	; Save position to set/clear ; Make it a multiple of 8 ; Save as a start for countins ; Divide by 8 ; Add offset to tab table ; Init pointer ; Get tab byte ; Loop through all 8 bits ; Get count and save byte ; If position if found ; Get value desired ; Place into tab byte ; Change back ; Rotate to next tab position ; Bump count ; Until all 8 bits done	
33 058F F4		MQV RET	@PTR1, A	; Replace the tab byte	

3 3 3 3 3 3

)

)

	1					7.01.		
	. 2		• • • • • • • • • • • • •			IAPT.	horizontal tab routin	€
	3					*TAB*	gets current position	and then does a search for the
	4 =				;	next :	tab stop. If none is	found then the default is the last
د	¦ ኒ	• • • • • • • • • • •	• • • • • • • • • • • • •	• • • • • • • • • •		colum	n.position	
	7				;			
8	3				***************************************	ENTRY	none	
·········	? ; · · · · · · · · ·							
11					; •	EXIT	vous	
12	2			• • • • • • • • • • • • • • • • • • • •		UŚEŚ	A, B, PTR1, WORK, T	
$\dots 13$							The second secon	ENT / ENEXT/E
12	 5 05C1	ce			77.0.50			
	0503 0503	E5 Fb	11		TAB:	MOV MOV	A, XPOS	; Get current position
	0504	54	F8			ANL	TEMP, A A, #11111000B	; Save position to tab from
is	7765C6	····Fċ····			• • • • • • • • • • • • • • • • • • • •	······································	WORK, A	; Make a multiple of 8 ; Save as a start for count
	0507	03				RR	A	Divide by 8
	0508 0509			,		RR	A	
	0307 056A	F5	Fö			RR MOV		
	0500	24	29			ADD	B, A A, #TABTAB	; Save temporary byte offset
	050E	F8			• • • • • • • • • • • • • • • • • • • •	พิธีจั้ง	PTRI, A	; Add offset to tab table ; Initialize pointer
	05CF	74	OA			MOV	A, #10	
	∵ö5bi∵ ′o5D3	95	FØ			SUBB	A, B	; Calc # times needed in Yoop
	05D3 05D4	FA C2	b5			MOV CLR	INDX1, A	; Place into index
29						CER	FO .	; FO ≈ hunting for current poD
	05b6	E9			TBL1:	······MÓV····	A, @PTR1	; REPEAT set tab byte
31	.05D7 ∵ö5bA…	30	D5 i7	02		JNB	FO, TB6	; IF not looking for position
	OSDA OSDC	60 7B	17 08		TD/a	Jż	TB1	s quick check to exit
34					TB6:	MOV	INDX2, #8	; look through the 8 bits ; REPEAT
35	05DE	20	D5	09	TBL2:	JB	FO, TB3	; REPEAN ; IF looking for current
	COSET	çb		• • • • • • • • • • • •		xch	A, TEMP	set original position
	.05E2 .05E5	B5 b2	04 D5	02			A, WORKA, TB4	; IF equal to original
	05E7	CD	DO		TB4:	XCH	FO	flag it
	05E8***	šõ	63			······ŠJMP···	A, TEMP TB2	; get current position ; ELSE looking for tab
	05EA	20	E7	OF	твз:	JB	ACC.7, TBR	; ELSE TOOKING for tab ; IF tab set THEN exit
	OSED	23			TB2:		<u>A</u>	rotate to next
	OSEE OSEF···	oc	ED			INC	WORK	; bump count
45		DB	ED				INDX2, TBC2	ONTIL bit count = 0
	`ö5F1```	89	·····6'4 · · · · ·	• • • • • • • • • • • • • • • • • • • •		····SJMP···	···TB5·····	; IF need to bump by eight
	05F3	74	08		TB1:	MOV	A, #8	, it has to bomb by 61aut
	`05F5```	2C			• • • • • • • • • • • • • • • • • • • •	ADD	A, WORK	5 bump count by eight
	05F6 05F7	FC 08			· · · · · · · · · · · · · · · · · · ·	MOV	WORK, A	
	05F8	DA	DC		TB5:	INC	PTRI	bump pointer to next byte
52			· · · · · .	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	ZNCO	INDX1, TBL1	; UNTIL table is done
	05FA	. 7C	4F			MOV	WORK, #MAXCHAR	; Default value if not found
54	OFFO						***************************************	
 ごご	.05FC .05FE	8C	<u>11</u>		TBR:	MOV	XPOS, WORK	; Set next tab position
	0600	21	A9			ACALL	BLINAD	; Build new address
		. <i>.</i> . 				JMP	SPF	; Skip any protected fields

1	,			5 5	BTAB -	horizontal back ta	ab routine
 2				5	*RTAR*	gets current posit	tion and then does a search for the
 3			• • • • • • • • • • • • • • • • • • • •		previo	us tab stop. If no	one is found then the default is the
5					first	column position	
 6				;			
 7					ENTRY	none	
9				· ,			
 10				•	EXIT	none	
 ${12}^{11}$					USES	A, B, PTR1, WORK	, TEMP, INDX1/2
				,			
 <u>13</u>						A MOOR	; Position to compare with
 15.0602	E5	11		BTAB:	MOY JZ	A, XPOS BTR	; IF first column THEN exit
	14	17			DEC	A	; Decrement for compare
 i8.0607	75 75	11			MOV	XPOS, #0	; Start looking from first pos
 . 19	ي				MOV	B, XPOS	; REPEAT save previous position
	85 ····	11 E0	FO	BTL:	PUSH	ACC	; save compare value
 22 060F	co ····	FÖ			PUSH	B	; save previous position
 23 0611	B1	<u>C</u> 1			ACALL	TAB	; tab to next stop ; restore previous position
	DO DO	FO EO			POP	ACC	; restore compare value
 25 0615 26 0617	B5	<u></u> 11	00		CUNE	A, XPOS, BTL1	
	50	EE		BTL1:	JŅÇ	BTL	; UNTIL next >= current Pos
 28	OF	: F0	11		MOV	XPOS, B	; Get previous position
29 061C	80						
 29 061C 30 061F	85 81	EA		BTR:	JMP	BLINAD	; Calc new address and return
 29 061C 30 061F	81	ÉÅ				BLINAD	; Calc new address and return
 29 061C 30 061F	65 81	EĂ				BLINAD	; Calc new address and return
29 061C 30 061F	81	EĂ				BLINAD	; Calc new address and return
29 061C 30 061F	81	EĂ				BLINAD	; Calc new address and return
29 061C 30 061F	81	EĂ				BLINAD	; Calc new address and return
29 061C 30 061F	81	EĂ				BLINAD	; Calc new address and return
29 061C 30 061F	81	EĂ				BLINAD	; Calc new address and return
29 061C 30 061F	81	EĂ				BLINAD	; Calc new address and return
29 061C 30 061F	81	EĂ				BLINAD	; Calc new address and return
29 061C 30 061F	81	EA		BTR:		BLINAD	; Calc new address and return
30 061F	81	EA		BTR:	JMP	BLINAD	; Calc new address and return
 30 061F	81	EA		BTR:	JMP	BLINAD	; Calc new address and return
 30 061F		EA		BTR:	JMP	BLINAD	; Calc new address and return
 30 061F		EA		BTR:	JMP	BLINAD	; Calc new address and return

)))))))

)

)

			HT\$.borizontal.tab.set	t	
		;		sets.tab.at.cunner		•••••
4 5		7		A 5. 1 F. 1 J. 1 J. 1 J. 1 J. 1 J. 1 J. 1 J	.C. FYRT.1809	
6		;	ENTRY	none		• • • • • • • • • • • • • • • • • • • •
/ 8	•••••	· · · · · · · · · · · · · · · · · · ·	EXIT	none		
9 10	•••••		USES		·····	
<u>11</u>		, 		all		
13 0621 D2	B5 11	HTS:	SETB MOV	FO	; Set tab	
15 0625 A1	11 A2	HTS1:	MOV AJMP	FO A, XPOS SETTAB	; Get position	
16 17					; Execute and return	
18		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		
19 20	• • • • • • • • • • • • • • • • • • • •	,	нтс – 1	 Horizontal tab cle		
21 22	•••••					
23 24		• •••••	*H1C*	Clears tab at curr	ent position.	
24 25 26		; ;	ENTRY	none		
26 27 28		7				• • • • • • • • • • • • • • • • • • • •
28 29	•••••	**************************************	EXIT			
30	• • • • • • • • • • • • • • • • • • • •		USES	all		
31 32 0627 - C2		HTC:	CLR	F0		
33 0629 80 3 4	F8		SJMP	HTS1	; Clear tab ; Execute and return	
35 36						
36 37 38				••••••		• • • • • • • • • •
39		**************************************	ATBC -	ANSI tabulation c	lear	• • • • • • • • • • • • • • • • • • • •
40	•••••		*ATBC*	performs the spec	ified tab controls.	
41 42						
43 44			ENTRY	none		
45 46		,	EXIT	none		
1 7		; ;	USES	all		
is 19		• • • • • • • • • • • • • • • • • • • •				
50 062B 112 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12 11 12	09D1	ATBC:	LCALL	GNP	; Get parameter	
51 062E 60 52 0630 B4	F7 03 02		JZ CUNE	HTC A, #3, AT2	; IF O THEN hor tab clear ; ELSE IF 3 THEN	
53 0633 A1 54 0635 22		AT2:	AJMP	CLRTABS	; ELSE IF 3 THEN ; clear all tabs ; ELSE return	
	***************	A12:	RET		; ELSE return	
				• • • • • • • • • • • • • • • • • • • •	••••••	

C C C C C C C C C C

ENTRY	1				; ;	HTAB -	Hazeltine tab routine	
ENTRY none EXIT none EXIT none Second of the properties of the						*#TAB*	performs the Hazeltine	tah function.
ENTRY	3 4					ждина»	Petrolina the hazertare	
EXIT	5					FNTRV		
	6 7				, ,		none	
1	8				*	EXIT	none	
1	.9 .0					··········ùšés···	all	
3 0636 C2 D5 HTAB: CLR FO FRISE FHLSE 4 0638 E5 11 MOV A, XPOS ; Check for home Position 5 063A 25 12 ADD A, YPOS ; Check for home Position 6 0 ADD A, YPOS ; Check for home Position 7 063C 70 02 JNZ HTO ; IF home Position THEN 8 063E D2 D5 SETB FO ; Flas = TRUE 9 0 0640 C0 11 HTO: PUSH XPOS ; Save current Position 1 0642 C0 12 PUSH YPOS ; Save current Position 1 0642 C0 12 PUSH YPOS ; Save current Position 1 0644 75 3F 80 MOV PFIELD; #10000000B ; Make normal Protected 3 0647 73 AP ACALL SPF ; Skip over foreground 4 0649 E5 11 MOV A, XPOS ; Scheck for home Position 7 064B 25 12 ADD A, YPOS ; Check for home Position 8 064F 30 D5 99 JNB FO, HT2 ; IF that is old POS THEN 9 064S 75 3F 08 HT1: MOV PFIELD; #00001000B ; Make half protected 1 065S 31 AP MOV A, XPOS ; Check for home Position 1 065S 31 AP MOV A, XPOS ; Check for home Position 1 065B 75 3F 00 HT2: MOV PFIELD; #00001000B ; Make half protected 1 065B 75 3F 00 HT2: MOV PFIELD; #00001000B ; Skip over backsround 1 065C 05 3F 05 JP MOV A, XPOS ; Check for home Position 1 065C 05 JP MOV A, XPOS ; Check for home Position 1 065C 05 JP MOV PFIELD; #0000100B ; IF POSITION THEN 1 065C 05 JP MOV PFIELD; #0000100B ; Skip over backsround 1 065C 05 JP MOV PFIELD; #0000100B ; IF POSITION THEN 1 065C 05 JP MOV PFIELD; #0000100B ; Skip over backsround 1 065C 05 JP MOV PFIELD; #0000100B ; IF POSITION THEN 1 065C 05 JP MOV PFIELD; #0000100B ; IF POSITION THEN 1 065C 05 JP MOV PFIELD; #0000100B ; MOV PFIELD; #000	1							
Sociation Soci		C2	D5		HTAB:		FO	; Flas = FALSE
No.	4 0638						A, XPOS	* Check for home position
7 063C 70 02 JNZ HTO	5.063A	25	12			AUU	A, YPUS	
9 0 0640 CO 11 HT0: PUSH XPOS Save current position 1 0642 CO 12 PUSH YPOS Save current position 1 0642 CO 12 PUSH YPOS Save current position 2 0644 75 SF SO MOV PFIELD, #1000000B Make normal protected 3 0647 75 SF SO ACALL SPF Skip over foreground 4 0649 E5 11 MOV A, XPOS Check for home position 5 0648 25 12 ADD A, YPOS Check for home position 6 0 0640 70 03 JNZ HT1 IF home position AND 8 064F 30 D5 09 JNB FO. HT2 IF that is old pos THEN 9 065C 31 A9 ACALL SPF Skip over background 1 065T 31 A9 ACALL SPF Skip over background 1 065T 5 11 MOV PFIELD, #00001000B Make half protected 1 065B 75 SF 00 HT2: MOV PFIELD, #0 Clear protected fields 1 065C 00 D		70	02			JNZ	нто	; IF home position THEN
O 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	8 063E	D2	D5			SETB	FO	; flag = TRUE
1 0642 C0 12 2 0644 75 3F 80 MOV PFIELD, #10000000B ; Make normal protected 3 0647 31 A9 ACALL SPF ; Skip over foreground 4 0649 E5 11 MOV A, XPOS 5 064B 25 12 ADD A, YPOS ; Check for home Position 6 ADD A, YPOS ; Check for home Position AND 8 064F 30 D5 09 JNB F0, HT2 ; IF home Position AND 8 064F 30 D5 09 JNB F0, HT2 ; IF that is old POS THEN 9 0652 75 3F 08 HT1: MOV PFIELD, #00001000B ; make half protected 10 0657 E5 11 MOV A, XPOS ; Check for home Position 11 0657 E5 11 MOV A, XPOS ; Check for home Position 12 0658 75 3F 00 HT2: MOV PFIELD, #0 1 Clear Protected fields 13 0658 75 3F 00 HT2: MOV PFIELD, #0 1 Clear Protected fields 14 065B 75 3F 00 HT2: MOV PFIELD, #0 1 Clear Protected fields 15 0660 DO E0 POP ACC ; and return 16 0665 DO 12 HT3: POP YPOS ; ELSE 17 0665 DO 12 HT3: POP YPOS ; Use Previous Position	9 '0 '0640'''	co	11		нто:	PUSH	XPOS	; Save current position
3 0.647 31 A9		CO				PUSH		
4 0649 E5 11				80				
## ADD A. YPOS		31						7 3817 3401 101031 3313
10								; Check for home position
8 064F 30 D5 09 JNB F0, HT2 ; IF that is old pos THEN 9 0652 75 3F 08 HT1: MOV PFIELD, #00001000B ; make half protected 0 0655 31 A9 ACALL SPF ; skip over background 1 0657 E5 11 MOV A, XPOS 2 0659 25 12 ADD A, YPOS ; check for home position 3 ADD A, YPOS ; check for home position 3 ADD A, YPOS ; Clear protected fields 5 ADD ACC ; remove old position 8 0662 D0 E0 POP ACC ; remove old position 8 0664 22 RET 0 0665 D0 12 HT3: POP YPOS ; ELSE 1 0665 D0 11 POP XPOS ; use previous position	6						1174	. IE home position AND
9 0652 75 3F 08 HT1: MOV PFIELD, #00001000B ; make half protected 0 0655 31 A9 ACALL SPF ; skip over background 1 0657 E5 11 MOV A, XPOS 2 0659 25 12 ADD A, YPOS ; check for home position 3 Clear protected fields 5 Clear protected fields 5 Clear protected fields 5 Clear protected fields 6 065E 60 05 JZ HT3 ; IF position not home THEN 6 066E 60 05 JZ HT3 ; IF position not home THEN 7 0660 DO EO POP ACC ; remove old position 8 0662 DO EO POP ACC ; and return 19 0664 22 RET 10 0665 DO 12 HT3: POP YPOS ; ELSE 12 0667 DO 11 POP XPOS ; use previous position	7 064D	79	<u>.03</u>					; IF that is old pos THEN
1 0657 E5 11 MOV A, XPOS 2 0659 25 12 ADD A, YPOS ; check for home position 3 4 065B 75 3F 00 HT2: MOV PFIELD, #0 ; Clear protected fields 5 6 065E 60 05 JZ HT3 ; IF position not home THEN 77 0660 D0 E0 POP ACC ; remove old position 88 0662 D0 E0 POP ACC ; and return 19 0664 22 RET 0 10 0665 D0 12 HT3: POP YPOS ; ELSE 12 0667 D0 11 POP XPOS ; use previous position					HT1:			; make half protected
2 0659 25 12 ADD A, YPOS ; check for home position 3	0 0655	31						; skip over background
3 4 065B 75 3F 00 HT2: MOV PFIELD, #0 ; Clear protected fields 5 6 065E 60 05 JZ HT3 ; IF position not home THEN 7 0660 D0 E0 POP ACC ; remove old position 8 0662 D0 E0 POP ACC ; and return 9 0664 22 RET 0 1 0665 D0 12 HT3: POP YPOS ; ELSE 2 0667 D0 11 POP XPOS ; use previous position	1 0657	E5					Α, XPUS Δ. VPNS	; check for home Position
MOV PFIELD, #0 Clear protected fields Fig. MOV PFIELD, #0 Clear protected fields Fig.								
10 10 10 10 10 10 10 10	34 065B	75	3F	00	HT2:	MOV	PFIELD, #0	; Clear protected fields
17 0660 DO EO POP ACC Femove old Position Ref RET RESE RE		AO	65			jż	нтз	; IF position not home THEN
RET 10 10 11 10 12 11 11 12 12 13 14 15 15 16 17 17 18 18 18 18 18 18 18 18 18 18 18 18 18	37 0660					POP	ACC	; remove old position
0 1 0665 DO 12 HT3: POP YPOS ; ELSE 12 0667 DO 11 POP XPOS ; use previous position	88 0662	DO	EO				ACC	; and return
1 0665 DO 12 HT3: POP YPOS ; ELSE 12 0667 DO 11 POP XPOS ; use previous position		??						
12 0667 DO 11 POP XPUS , use Previous Position		DO	12		нта:			
	12 0667 13 0669	DO 81	11 EA			POP AJMP	XPOS BLINAD	; use previous position ; update and return
	• • • • • • • • • • • • • •							
\cdot			• • • • • • • • • • • •					

***.Z-	•							
	12		•••••		RING -	rina bell		
	3				*RING*	rinas the bell on th	he keyboard. If the input FIFO	***************************************
	5			; ;	has a l	bell as the next char t the terminal can ke	racter it is dropped. This is eep up with multiple bells and	
	6 7			; ;	commun:	icate to the keyboard	d.	•••••••••••
	8 9	********	••••••	*			••••••	•••••
1	10		••••••		ENTRY			
1:	11 12	• • • • • • • • • • • • • • • • • • • •	•••••	5	EXIT	none		•••••
		• • • • • • • • • • • • • • • • • • • •			USES	none	***************************************	•••••
1	.5			····				
	.6 066B 74 7 066D 51	81 6D		RING:	MOV ACALL	A, #KBC_BELL PCKB	; Output command to ring be	-11
18 19	.8	1B	************	RGL:	ACALL	LCIF		
20	0 0671 40	07		.nor	JC	RGR	; REPEAT get next input cha ; IF no character THEN ex	×it
22	2 0676 31	97 F6		• • • • • • • • • • • • • • • • • • • •	CUNE			EN
23	3.947880	E5						
24	4 067A 22			RGR:	SUMP RET	RGL	; UNTIL char <> bell	
24	4 067A 22					. Not.	; UNTIL char <> bell	
24	4 067A 22					. NQL	; UNTIL char <> bell	
24	4 067A 22					. Not.	; UNTIL char <> bell	
24	4 067A 22					. Not.	; UNTIL char (> bell	
24	4 067A 22					. Not.	; UNTIL char <> bell	
24	4 067A 22					. Not.	; UNTIL char <> bell	
24	4 067A 22					. Not.	; UNTIL char (> bell	
24	4 067A 22					. Not.	; UNTIL char ⇔ bell	
24	4 067A 22					. Not.	; UNTIL char ⇔ bell	
24	4 067A 22			RGR:	RET	. Not.		
24	4 067A 22			RGR:	RET			
24	4 067A 22			RGR:	RET			
24	4 067A 22			RGR:	RET			

C

C

C

C C

54: 4 ?	COMPO	ER TERM	iiwar Fir	diwhire ' Xe	RSION 1.03 ***		
1					₹ ₹	ESCCODE	- escape code received
2					; :	*ESCCOI	E* determines which mode the terminal is in and
<u>3</u> .						sets th	e dispatch address to the proper routine.
5.							
<u>ž</u> .						ENTRY	none
9 10					, ;	EXIT	none
10					; ;	USES	none
12							
13.	067B	12	102D		ESCCODE:	LCALL	R2_ESC ; Escape hook
15	 067E	75	16	97		······································	DSADRL, #LOW (ANSIESC) ; ANSI dispatch address
	0681	75 75	15	ó6		MOV	DSADRH, #HIGH (ANSIESC)
18 19						IFC	NE, HIGH(ANSIESC)-HIGH(ZDSESC)
20						ERROR ENDC	\$ ZDSESCY IS NOT THE SAME AS ANSIESCY
21 22						ifc	NE, HIGH (ANSIESC) -HIGH (ADMSESC)
23 24						ERROR ENDC	; ADM3ESC / IS NOT THE SAME AS /ANSIESC/
25 26						IFC	NE,HIGH(ANSIESC)-HIGH(HAZESC) ;/HAZESC/ IS NOT THE SAME AS /ANSIESC/
26 27						ERROR ENDC	1. HATERC, IS NOT THE SHIPE HS HINSTERS
28							ANSI, ESCRET ; IF not ANSI mode THEN
	068 4 0687	20 75	40 16	OF B8		JB MOV	DSADRL, #LOW (ZDSESC) ; ZDS dispatch address
31	068A	20	41	09		JB	ZDS, ESCRET ; IF not ZDS mode THEN
33	068D		16	A7		MOV	DSADRL, #LOW (ADM3ESC) ; ADM 3 dispatch address
34	.0690	20	42	03		JB.	ADM3, ESCRET ; IF not ADM 3 mode THEN
36	0693	20 75	42 16	03 	ESCRET:	MOV RET	DSADRL, #LOW (HAZESC) ; Hazeltine dispatch address ; return
3.7.	. 0696	22			ESONE.		

Z29R0	M; MÍCRO	BENCH 80	51 CROSS AS	SEMBLER (V1)-218 24 ARE.VERSION.1.03.***	-FEB-83 15	:30:06 PAGE 58	
		OIEM. LEN	WITHHE . IS TELIM				······
	2	• • • • • • • • • • • • • • • • • • • •			ANSIES	CANSI.escape.pa	user
	3 4	• • • • • • • • • • • • • • • • • • • •			ANSIE	SC*.marses_the.fir ANSI_mode.	st.character.followins.an.ESC.when
	5					• • • • • • • • • • • • • • • • • • • •	
	. 7		• • • • • • • • • • • • • • • • • • • •		ENTRY	(A)=chanac.ten.	to.Parse
	.2 10			······	EXIT	DQD&	
	11			,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	USES	all	
	12 13						
	14 0697 15 0699	C2 Ç2	18 19	ANSIESC:	CLR CLR	PSDF	; Flas first time through *PSD*
	16 069B	75	F0 (DC	MOV	B, #ANSITL	
	17.,949E 18	90	97.98		MQV	DPTR:#ANSII	
	19.06A1 20 06A4	12 02	082F 0836	AESC1:	LÇALL.	SETNORM	
	21				LOMP		; Jump to any routine
	22 23						
	2 4 25				∆DM3E¢	C - ADM 2 ascana m	arser
3	26	• • • • • • • • • • •		;			
2	27 28	,	••••••		routin	e, if any, for ADM	
	29	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		•••••		
	31 32	• • • • • • • • • • • • • • • • • • • •			ENTRY	(A)=finstchar	acter.after.ESC.sequence
			•••••		EXIT	none	
	35		• • • • • • • • • • • • • • • • • • • •	; ,	USES	al.1	
	36 37						
3	38 06A7 39	12	082F	ADM3ESC:	LCALL	SETNORM	Restore dispatch address
4	10 06AA 11 06AD	B4 02	3D 3	1	CJNE	A, #'=', ZE2	; IF not '=' THEN return
	··• · · · · · · · · · · · · · · · · · ·		ХҮРҮ	••••	LJMP	Ç9 A	J. ELSE same as cursor address.
	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		
		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •				••••••
,			• • • • • • • • • • • • • • • • • • • •			••••	
				••••••••••			
***********				• • • • • • • • • • • • • • • • • • • •		•••••	
• • • • • • • • • • • • • • • • • • • •	•••••			•••••			
			•••••	•••••			
		•••••	• • • • • • • • • • • • • • • • • • • •	•••••			••••••
		• • • • • • • • • • • • • • • • • • • •		•••••			

C

(

C

Z29ROM;	MICROBE	NCH 8051	CROSS A	ASSEMBLER	(V1)-218 SION 1.03	24-FEB- ***	83 15:	30:06 PAGE 5	9							
• • • • • • • • • • • • • • • • • • • •		-:\! !\!!			atanatan !!		IAZESC ·	- Hazeltine	1500 escape	e (tilde) parser					
		,											er			
3						······	outine	* parses the , if any, fo	r Hazeltin	e 1500 m	ode.		Francis			
5 6							NTDV	(A) = first	chanacter	after F	sc (+ild	e) seque	 nce			
									Character.	artet	49YX#9	5.(75.79.5	1047		• • • • • • • • • • • • • • • • • • • •	
<u>9</u>								none						•••••		
11						u	ISES	.all								• • • • • • • • •
13 14	06B0	75	FO	io	HAZESC:		 10V	B, #HAZTL			eltine t		ameters			• • • • • • • • • • • • • • • • • • • •
15 16	06B3 06B6	90 80	0768 E9			s	10V SJMP	DPTR, #HAZT AESC1		; Loo	k throug	h table				
							•••••									
						• • • • • • • • • • • • • • • • • • • •										
													• • • • • • • • • •			
												• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			
													• • • • • • • • • • • • • • • • • • • •			
																• • • • • • • • • • • • • • • • • • • •
																• • • • • • • • •
																• • • • • • • • • • • • • • • • • • • •
																• • • • • • • •
														.		
															•••••	

•		RE. MERSION 1.03 **			••••••
··	•••••••		ZDSESC	- ZDS escape sequenc	<u> </u>
3			*ZDSES	C* parses the charact	ter received to do the proper ZDS
4 =		;	routin	e if any.	
6	••••••		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
7	• • • • • • • • • • • • • • • • • • • •	,	ENTRY	(A) = first charact	ter after ESC sequence
8		,			
10	• • • • • • • • • • • • • • • • • • • •		EXIT	none	
11			USES	all	
12					
13 14 06B8 12	082F	ZDSESC:	LCALL	SETNORM	
15	4		LONCE	SETNORM	; Restore dispatch address
16 06BB B4 17 06BE 02	23 03	3	CUNE	A, #'#', ZE1A	; IF transmit page THEN
18 06C1 B4	1036 2D 02	ZE1A:	JMP CJNÉ	R2_TP A, #'-', ZE1B	; do it
19 06C4 C1	02		AJMP	BTAB	; IF back tab THEN ; do it
20 06C6 B4 21 06C9 02	2E 03	ZE1B:	CUNE	A, #/./, ZE1C	; IF tab functions THEN
2 1 06C9 02	OED3	• • • • • • • • • • • • • • • • • • • •	JMP	ZTAB	; do it
23 06CC	• • • • • • • • • • • • • • • • • • • •	ZE1C:	CLR	С	; Set up to
	3C		SÚBB	C A, ¥2<2	; Normalize into range
25 06CF 40 26	OD	• • • • • • • • • • • • • • • • • • • •	JC	ZE2	; IF out of range THEN return
27 06D1 23 28 06D2 FD	• • • • • • • • • • • • • • • • • • • •		RL	A	; ELSE mult by 2
28 06D2 FD 29 06D3 24	00		MOV	TEMP, A	save value
29 06D3 24 30 06D5 83	OA	• • • • • • • • • • • • • • • • • • • •	ADD MOVC	A, #10 A, @A+PC	; add code offset to low
31 06D6 CO	E0		PUSH	ACC	<pre></pre>
32 06D8 ED 33 06D9 24	03		MOV	A, TEMP	; get value back
33 06D9 24 34 06DB 83	03		ADD MÖVC	A, #3 A, @A+PC	; add code offset to high
35 06DC CO	E0	••••	PUSH	ACC	; get high byte ; put onto stack
36 06DE 22		ZE2:	RET	••••••	; jump to address
37 38∵⋯⋯⋯⋯	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	······ifc·····	NE,ZDST-\$	
39 	• • • • • • • • • • • • • • • • • • • •		ERROR	; TABLE 'ZDST' MUST	FOLLOW
40			ENDC	***************************************	
•••••	••••••	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		
	• • • • • • • • • • • • • • • • • • • •				
			,		
• • • • • • • • • • • • • • • • • • • •	•••••		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	••••••	
	• • • • • • • • • • • • • • • • • • • •				

C

C

C

C

	IRMWARE VERSION 1.03 ***			
		ZDST -	ZDS escape table	
2	; :	*ZDST*	contains the seco	nd character in the range of
4		all es	cape sequences the	Z-29 will respond to.
5	,			······v·orresorvesseringspile 462
6	5	Table	entries are: Addr	ess of routine which performs the rder of their ASCII sequence.
7		. i.e.ues	ited ore: attom in o	!.M\$1XI\$1055117555357115333554
9				
10 06DF 0B03	ZDST:	DW	EAM	; < enter ANSI mode
11 06E1 OB4E		DW	EKAM	; = enter alt keypad mode ; > exit alt keypad mode
12 06E3 0B51 13 06E5 0767		DW	UNDEF	; ? undefined
14 06E3 0767 14 06E7 0B54		DW	EICM	; @ enter insert char mode
15 06E9 OC8D		D.W	CUP	; A cursor up
16 06EB OCA8		DW	CDN CRT	; B cursor down ; C cursor riaht
17 06ED 0C64		DW	CLFT	; D cursor left
18 06EF OC3D 19 06F1 OBAC		DW	CLRS	; E clear entire display
20 06F3 0F39		DW	EGM	; F enter sraphic mode
21 06F5 0F3C		W	XGM	; G exit graphic mode ; H set cursor home
22 06F7 0C36		DW	SCH PRLF	; H set cursor home ; I reverse index
23_06F90518		DW	EED	; J erase cursor to EOS
24 06FB 0C08 25 06FD 0BE0		. DW	EOL	; K erase cursor to EOL
26 O6FF OD47		DW	PIL	; L perform insert line
27 0701 OD88			PDL	; M perform delete line ; N delete char
28 0703 ODF4		DW	PDC XICM	; N delete char ; O exit insert char mode
29 0705 0B58 30 0707 0767		DW	A.P.D	; P undefined (blue)
30 0707 0767 31 0709 0767		DW	UNDEF	; Q undefined (red)
32 070B 0767		DW	UNDEF	; R undefined (gray)
33 070D 0767		D.W	UNDEF	; S_ undefined (F1)
34 070F 0767		DW DW	UNDEF UNDEF	; T undefined (F2) ; U undefined (F3)
35.07110767 36.07130767		P.W	UNDEF	; V undefined (F4)
36 0713 0767 37 0715 0767		DW	UNDEF	; W undefined (F5)
38 0717 0767		DW	UNDEF	; X undefined
39.0719QCD6		DW	<u>CUA</u>	; Y VT52 cursor addr (Pacam); Z VT52 identify
40 071B OEAB		DW DW	IDT EH\$M	; [Enter hold screen mode
41.071DQB3C 42.071F0B43		₽.W	XHSM	: \ Fxit hold screen mode
42 071F 0843 43 0721 1039		DW	R2_T25L	; 1 ×mit 25th line
44 0723 1033		DW	R2_TL	; ^ transmit line
45.07251030		<u>DW</u>	R2_TC R2_PRNT	: _ transmit character
46 0727 103C		DW DW	UNDEF	; a undefined
47 0729 0767 48 072B 0BF1		₽₩ DW	EBD	; b erase display to cursor
49 072B 05F1 49 072D 9767		DW	UNDEF	; c undefined
50 072F 0767		DW	UNDEF	; d undefined
51 0731 0767		<u>D</u> W	UNDEF	; e undefined ; f undefined
52 0733 0767		₽₩	UNDEF UNDEF	; r underined ; g undefined
53 0735 0767 54 0737 0767		₽₩ DW	UNDEF	; h undefined
54 0737 0767 55 0739 0EB3		DW	IDTT	; i identify terminal type
56 073B 0EE8		DW	SCP	; j save cursor position

.....

Z29ROM; MICROBENCH 8051 CROSS AS: ***.Z-28.COMPUTER.TERMINAL.FIRMW	SEMBLER (V1)-218 24 ARE.VERSION 1.03 ***	4-FEB-83 15 *	***************************************	
1.073FOBDB		DW	EEL	
2 0741 0767	• • • • • • • • • • • • • • • • • • • •	DM	UNDEF	; l erase entire line
3 0743 OE83		DW	CPR	; m undefined
4 0745 OBCB		DW	EBL	n cursor position report
		DW		; o erase line to cursor
6 07 4 9 0F36	• • • • • • • • • • • • • • • • • • • •	DW	XRVM	P. enter reverse video mode
				; 9 exit reverse video mode
8 0 74 D 0F08		DW	ZATR	
9.074FQB48		DW	EKSM	; s set video attributes
10 0751 OB4B	***************************************	DW.	XKSM	; t enter keypad shifted mode
11. 0753 QAFD		DW	WEOL	; u exit kerpad shifted mode ; v enter wrap around mode
12 0755 OBOO		DW	DEOL	
13.075708 <u>E</u> 7		DIW		
14 0759 092C		DW	RMS	
15. 075B QAFA		DW	RAMP	; Y reset modes (param)
16 075D OFB3		DW	EKI	; z reset to power up ; (enable keyboard
17. 075F 0767		DW		· I compared and
18 0761 OFB9			DKI	; ! undefined ; } disable keyboard
19.07630767				e au comunicación de descripción de
20 0765 0767		DW	UNDEF	
21				, pcr undefined
22 23				
24			***************************************	
26		UNDEF	T.undefined.escape	sequence
		ALIMITIEE.	x := := 700 .	
27 28		*UNDEF	* is used in ZDS e	scape parsing for undefined entries
28				scape parsing for undefined entries
28 29 30	······································	.,		scape parsing for undefined entries
28 29				
28 29 30		ENTRY	none	
28 29 30 31		ENTRY EXIT	none	
28 29 31 32 33 34	5 5 7 7	ENTRY EXIT	none	
28 	5 5 7 7 7	ENTRY EXIT USES	none none none	
28 27 30 31 32 33 34 35 36	5 5 7 7 7	ENTRY EXIT USES	none none none	
28 27 30 31 32 33 34 35	5 5 7 7 7	ENTRY EXIT USES	none none none	
28 29 31 32 33 34 35	5 5 7 7 7	ENTRY EXIT USES	none none none	
28 27 30 31 32 33 34 35	5 5 7 7 7	ENTRY EXIT USES	none none none	
28 27 30 31 32 33 34 35	5 5 7 7 7	ENTRY EXIT USES	none none none	
28 29 30 31 32 33 34 35 36 37,0767 22	; ; ; ; ; ; ; UNDEF:	ENTRY EXIT USES RET	none none none	; Undefined ESC sequence
28 29 30 31 32 33 34 35 36 37,0767 22	; ; ; ; ; ; ; UNDEF:	ENTRY EXIT USES RET	none none none	; Undefined ESC sequence
28 29 30 31 32 33 34 35 36 37, 9767 22	; ; ; ; ; ; ; ; ; ; ; ; ;	ENTRY EXIT USES RET	none none none	; Undefined ESC sequence
28 29 30 31 32 33 34 35 36 37, 9767 22	; ; ; ; ; ; ; ; ; ; ; ; ;	ENTRY EXIT USES RET	none none none	; Undefined ESC sequence
28 27 30 31 32 33 34 35 36 37, 9767 22	; ; ; ; ; ; ; ; ; ; ; ; ;	ENTRY EXIT USES RET	none none none	; Undefined ESC sequence
28 29 30 31 32 33 34 35 36 37, 9767 22	; ; ; ; ; ; ; ; ; ; ; ; ;	ENTRY EXIT USES RET	none none none	; Undefined ESC sequence
28 27 30 31 32 33 34 35 36 37, 9767 22	; ; ; ; ; ; ; ; ; ; ; ; ;	ENTRY EXIT USES RET	none none none	; Undefined ESC sequence
28 27 30 31 32 33 34 35 36 37, 9767 22	; ; ; ; ; ; ; ; ; ; ; ; ;	ENTRY EXIT USES RET	none none none	; Undefined ESC sequence
28 27 30 31 32 33 34 35 36 37, 9767 22	; ; ; ; ; ; ; ; ; ; ; ; ;	ENTRY EXIT USES RET	none none none	; Undefined ESC sequence
28 27 30 31 32 33 34 35 36 37.9767 22	; ; ; ; ; ; ; ; ; ; ; ; ;	ENTRY EXIT USES RET	none none none	; Undefined ESC sequence
28 27 30 31 32 33 34 35 36 37.9767 22	; ; ; ; ; ; ; UNDEF:	ENTRY EXIT USES RET	none none none	; Undefined ESC sequence
28 27 30 31 32 33 34 35 36 37, 9767 22	; ; ; ; ; ; ; UNDEF:	ENTRY EXIT USES RET	none none none	; Undefined ESC sequence
28 29 30 31 32 33 34 35 36 37,0767 22	; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	ENTRY EXIT USES RET	none none	; Undefined ESC sequence
28 29 30 31 32 33 34 35 36 37, 0767 22	; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	ENTRY EXIT USES RET	none none	; Undefined ESC sequence
28 29 30 31 32 33 34 35 36 37, 0767 22	; ; ; ; ; UNDEF:	ENTRY EXIT USES RET	none none none	; Undefined ESC sequence
28 29 30 31 32 33 34 35 36 37, 0767 22	; ; ; ; ; UNDEF:	ENTRY EXIT USES RET	none none none	; Undefined ESC sequence
28 29 30 31 32 33 34 35 36 37,0767 22	; ; ; ; ; ; ; ; ; ; ; ; ;	ENTRY EXIT USES RET	none none	; Undefined ESC sequence
28 27 30 31 32 33 34 35 36 37 0767 22	; ; ; ; ; ; ; ; ; ; ; ; ;	ENTRY EXIT USES RET	none none	; Undefined ESC sequence
28 29 30 31 32 33 34 35 36 37 9767 22	; ; ; ; ; ; UNDEF:	ENTRY EXIT USES RET	none none none	; Undefined ESC sequence

C

C C

1			HAZT -	- Hazeltine 1500	escape (tilde) table
2 3 4		; ;	*HAZT:	* contains the se e (tilde) table.	cond character of the Hazeltine
5 6 7 8		; ; ;	Table escar which	entries are: The (tilde) sequence performs the req	e second character of the Hazeltine e followed by the address of the routine wested function.
9 0 1	0768	HAZT	EQU	\$	
2 07/0			DB	12H	: ~ DC2
3 0768 4 0769	. 12 . 0036		DW	sch	; ~ DC2 ; Set cursor home
5					
6 076B	OC.		DM DB	OCH CUP	
7 076C 8	OCSD			CUT	; Cursor up
9 076E	ов		DM DB	овн	; ~ VT ; Cursor down
0 076F	OCA8		DW	CDN	; Cursor down
21 22 0771 · · ·			DB	ich	; ~ FS
22 0771 23 0772	OBAC		DW	CLRS	; Clear screen and home
24					
25 077 4 26 0775	1D OBC2		DB	1DH CLFO	; ~ GS ; Clear foreground
26 0775				CLFO	
27 28 0777	OF		DB	OFH	; ~ SI
29 0778	OBEO		DW	EOL	; Erase to end of line
30	40		np	100	: ~ CAN
31 077A 32 077B	18 ocos		DB	18H EED	; ~ CAN ; Erase to end of display
33 34 077D					
4 077D	17		DB DB	17H HEED	; ~ ETB ; Erase to eod (with bckgrnd)
85 077E	OC19				, crase to end that beksing,
37 0780	15		DB DW	15H DKI	; ~ NAK ; Disable keyboard input
37 0780 38 0781	15 OFB9		DM	DKI	; Disable keyboard input
39 10 0783			DB	06Н	; ~ ACK
10 0783 11 0784	06 of.B3		DW	EK I	; Enable keyboard input
12					
13 0786 14 0787	1A 0B47		DB DW	1AH PIL	; ~ SUB ; Perform insert line
14 0787 15					
6 0789	i3		DB	13H	; ~ DC3
7 078A	OD88		DW	PDL	; Perform delete line
18			nn	100	: ~ FM
19 078C 30 078D	19 0F2A		DB DW	19H EHALF	; ~ EM ; Enter half intensity
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	VI 40				
52 078F	1F		DB	1FH	; ~ US
53.0790	9F26		DW	XATR	; Exit attributes
5 4 55 , 9792	11		pB	11H	; ~ DC1 <col/> <row></row>
№.× 4<br 56 0793	OCD6		ĎŴ	CUA	; Cursor address
57					

	1 0795	05			DB	05H	: ~ ENO	
	2	05 0E83			DB DW	CPR	; Cursor	Position report
	4	0010	•••••••	HAZTL	EQU	(\$-HAZT)/3	; Table	length
	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •					
		• • • • • • • • • • • • • • • • • • • •						
		•••••						
				•••••••				
*************		• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	•••••	• • • • • • • • • • • • • • • • • • • •	
• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	•••••					
						••••		
,	• • • • • • • • • • • • • • •	,						
•							•••••	
••••••		•••••		• • • • • • • • • • • • • • • • • • • •		••••••	••••••••	
• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •				
							• • • • • • • • • • • • • • • • • • • •	
	• • • • • • • • • • • • • • • • • • • •							
	• • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	••••••	
• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •				
			•••••					
		•						
	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	••••••	•••••••••••••••••••••••••••••••••••••••
	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •			
		• • • • • • • • • • • • • • • • • • • •						
		••••••••••••	• • • • • • • • • • • • • • • • • • • •		•••••	• • • • • • • • • • • • • • • • • • • •		••••••
	• • • • • • • • • • • • • • • • • • • •	••••••	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		•••••	
,								
		• • • • • • • • • • • • • • • • • • • •						
• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	•••••			•••••••••••••••••••••••••••••••••••••••
• • • • • • • • • • • • • • • • • • • •		•••••	·····			•••••	• • • • • • • • • • • • • • • • • • • •	

C

	; ;	ANSIT	- ANSI escape table	
	;	*ANSI	T* contains the seco	nd character of the ANSI escape table
	\$ \$ \$ \$	sequei	entries are: The sonce followed by the equested function.	econd character of the ANSI escape address of the routine which performs
0798	ANSIT	EģŪ	\$	
0798 5B		DB	<u>r</u> t	; ESC [
0799 07BC		DW	ELB	; Escape left bracket
079B 28		DB DW	GODES	; ESC (; GO designator
079C 0F3F				
079E 29		DB	G1DES	; ESC) ; G1 designator
079F 0F47		DW		
07A1 37 07A2 0EE8		DB DW		; ESC 7 ; Save cursor position
O/A2 OEE8				
07A4 38		DM DB	/8/ USCP	; ESC 8 ; Unsave cursor position
07A7 3D 07A8 0B4E		DB DW	(=/ EKAM	; ESC = ; Enter keypad alternate mode
		DB	:	; ESC >
07AA 3E 07AB 0B51		DM DR	XKAM	; Exit keypad alternate mode
				: FSC M
Q7AD4D		DW DB		; ESC M ; Reverse index (rev line feed)
		DB		; ESC c
07B0 63 07B1 0AF6		DW	ARAMP	; ANSI reset to power up
07B3 44		DB	^D^	; ESC D
07B4 0561		DB DW	^D^ PLF	; ESC D ; Perform line feed
		DB	HZ	; ESC H
07.B7 06.21		DW	HTS	; Horizontal tab set
.07.B923		DB DW		; ESC # 7 ; Print page
07BA OADF		DW	APRNT	; Print page
000C	ANSITL	EQU	(\$-ANSIT)/3	; Table length

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-21	8 24-FEB-83 15:30:06 PAGE 66	······································
.7-29.COMPUTER.TERMINAL.FIRMWARE.YERSION_1.0	g. ELB – escape left bracket	······
2 ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	*ELB* is a continuation o for ANSI escape sequence sequence is decoded with	f the escape sequence processing The final character of the or without a preceeding parameter sed on to the associated routine.
8 , ,	ENTRY none	
10 ; 	EXIT to requested rout	ine
13 14	USES all	
15 16 07BC 12 082A ELB:	LCALL SETDISP	; Set dispatch and return
17 18 07BF 12 0971 19 07C2 50 78 20 07C4 12 082F 21 07C7 75 F0 1E 22 07CA 90 07D0 23 07CD 02 0836	LCALL PSD JNC STRET LCALL SETNORM MOV B, #ELBTL MOV DPTR, #ELBT LJMP STJMP	; Get parameter string ; IF not done THEN return ; ELSE restore dispatch address ; (B) = table length ; (DPTR) = ESC [table ; Jump to any routine
		3. Same 10. 803. (Add 1)16.
		······································
		•••••••••••••••••••••••••••••••••••••••
		······
		······································
	•••••••••••••••••••••••••••••••••••••••	•••••••••••••••••••••••••••••••••••••••
	••••••	······································
	••••••	
		······

		; ;	ELBT	- escape left bra	cket table
		* 	*ELBT	* contains the th	ird and/or final character of the
		; ;			
		*	Table	entries are: Fi	nal character of ESC E Pn F sequence the routine which performs the
			reque	sted function.	
	07D0	ELBT	EQU	\$	
07D0	3E		DB	/3/	; ESC [> Ps F
07D1	0898	•••••	DW	/>/ A2M	; ESC [> Ps F ; ANSI set mode #2
3 3			DB		; ESC [? Ps F
07D4	0880		DW	A1M	; ANSI set mode #1
07D6	41		DB	ACUP	; ESC [Pn A ; ANSI cursor up
07D7	41 0C9D			ACUP	; ANSI cursor up
	42		DB	By	; ESC [Pn B
07DA	OCBB		DW	ACDN	; ANSI cursor down
O7DC	43		DB	/C/ ACRT	; ESC [Pn C ; ANSI cursor right
07DD	OCSE.				
07DF	44		DB	/D/	; ESC [Pn D
	OC46		DW	ACLFT	; ANSI cursor left
07E2 07E3	48		DB	'H.' APCA	; ESC [Pn ; Pn H ; ANSI perform cursor address
	09FA				
07E5	4A		DB	,၂,	; ESC [Ps J
. 97E4	0A1A		DW	EID	; Erase in display
07E8	4B		DB	′K′	; ESC [Ps K ; Erase in line
07E9	0A2B				
07EB	4C		DB	APIL	; ESC [Pn L ; ANSI perform insert line
97.E C	OD7D		DW		
07EE	4D		DB	´M´ APDL	; ESC [Pn M ; ANSI perform delete line
07EF	ODB6				
07F1	50		DB DW	P/ APDC	; ESC [Pn P ; ANSI perform delete char
97F2					
07F4	5A		DB DW	´Z.´ APBT	; ESC [Pn Z ; ANSI perform back tab
07F5	0006		ПМ	HFD1	
07F7	63		DB DW	^c^ APA	; ESC [Pn c ; ANSI device attributes
€.97F8 }	OEC3				
0.07FA	66		DB	f.′ APCA	j ESC [Pn : Pn f

.....

1 07FD	67		DD	/-/	
2 07FE	67 062B	• • • • • • • • • • • • • • • • • • • •	DB DW	9.′ ATBC	; ESC [Ps g ; ANSI tabulation clear
				A L D C	, HWST (ADDIACTOR CLEAR
4 0800	68		DB	'h'	; ESC [Ps h
5.0801.	Q83D		₽₩	A\$M	
	60		pp.		
8 0804	6C 085F	• • • • • • • • • • • • • • • • • • • •	DB DW	<u>(1</u>	; ESC [Ps] ; ANSI reset mode
			2**	rat N 1) MNST Lezet mode
10 0806	6D		DB	/m/	; ESC [Ps m
11 0807 12	OA3B		DW	ASGM	: ANSI set araphics mode
13 0809	6E		DB	(n/	· FOC F D
14 080A	GE OAAF	• • • • • • • • • • • • • • • • • • • •	ĎŴ	ACPR	; ESC [Pn n ; ANSI cursor position report
15 16 0800	···· <u>·</u> ····				, 11101 carson rosition report
16 080C	70		DB	'P'	; ESC [P
17 080D	1030		DW	R2_TC	; Transmit page
19 080F	72		DB	, KrK	; ESC [Pn ; Pn r
20 0810	72 0D24		ĎŴ	(rí DSR	; ANSI define scrolling region
21 22 0812	7 3				
22 0812	73 OEEF		DB	5'	; ESC [s
24	VEEF		DW	ASCP	; ANSI save cursor position
25 0815 26 0816	75 0F03		DB	~u~	; ESC [u
26 0816	0F03	• • • • • • • • • • • • • • • • • • • •	DB	AÜSCP	; ANSI unsave cursor position
27 28 0818	 76		<u></u>		
29 0819	0F96		DB DW	CDLD	; ESC [Pn v
30		• • • • • • • • • • • • • • • • • • • •	bw	SBLR	; Set blink rate
31 081B	77 OAÉA		DB	′ω′	; ESC [Pn w
32 0810 33	OAEA		bW	ASBR	; ANSI set baud rate
34081E.	·····7A	• • • • • • • • • • • • • • • • • • • •	ĎB	, z ,	; ESC [Pn z
35 081F	OAF6		DW	ARAMP	; ANSI reset to power up
36					
37 0821 38 0822	7B 0B7A		DB DW		; ESC [Ps {
39			ΠM	PCLK	; Program clock
39 40 0824			DB	,	; ESC [Ps)
41 0825	0A65		DW	ASPF	; Set protected fields
42 43 0827	40				
44 0828	ODE9		DB		; ESC [Pn @
45				HLIC.	; ANSI perform insert character
46	001E	ELBTL	ĔĠÜ	(\$-ELBT)/3	; Table length
				• • • • • • • • • • • • • • • • • • • •	
*******************	·····	•••••••••		• • • • • • • • • • • • • • • • • • • •	
		• • • • • • • • • • • • • • • • • • • •			••••••
** * * * * * * * * * * * * * * * * * * *	••••••		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
		***********			••••••
				· · · · · · · · · · · · · · · · · · ·	

1	TER TERM			. ;;) — set dispatch a	ddress
<u>1</u>	, 			# #	xerrnie	P* ests the dispa	tch address to the address just
4					followi	ing the call to th	is routine. This routine then refull with this. A call to this
5				.	routine	does not return	to back to the routine that called
					this or	ie.	
9				, ,			
10				, , , , , , , , , , , , , , , , , , ,	ENTRY	none	
. <u>11</u>					EXIT	none	
13 14					··········ÚŠÉŠ····	none	
15							
16 17 082A	DO	15		SETDISP:	POP POP	DSADRH	; Get high byte of data
18 082C	DO	<u>15</u> 			POP RET	DSADRL	; Get low byte of data
.19 082E 20	22						
.21 .22							
. 23 . 24							
24 25				; ; ;		M — set dispatch n	
.25 .26					*SETNOI	RM* sets the dispa o the 'normal' add	tch address for processing characters
.27 28					pack.c	oiieiioi.iiiaia.aa	V. 9.8.7.V.
 30					ENTRY	none	
30 31 32							
32				; ;	EXIT	none	
33 34				***************************************	USES	none	
. 35 							
.37.082F 38.0832	75 75	<u>1.6</u>	87	SETNORM:	MOV MOV	DSADRL, #LOW (NO DSADRH, #HIGH (N	DRM) ; Restore dispatch address
38 0832 39 0835	22	15			RET	DOMBINI WILLOW VI	

)))))))))

1								
2	• • • • • • • • • • • • • • • • • • • •	*******	• • • • • • • • • • • • • • • • • • • •	; ;	STCALL - jum	ch.table.and P to routine	.jump	
			• • • • • • • • • • • • • • • • • • • •	· · · · · · · · · · · · · · · · · · ·			ble operation. If the data was f	•••••
5.					itdoes.aiu	metothe.ad	dress in the (DPTR), otherwise it	ound
6 				, -	does a retur	n.		
8 				3	*SICALL only	does a jump	to the address in the (DPTR).	
10			•••••	,	••••••••		•••••	
11.		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		ENTRY(DP:T	R)=addness	.ta.jume.ta	• • • • • • • • • • • • • • • • • • • •
43				, 				
14 15.				; •				
16			•••••					
	0836	31	F7	ST. IMP:	ACALL XSTA	ъ		
12	.0838	40	92		JCSTRE	T	; Search the table 	turn
20 21.		E4						
23	083B	73			OMP CA+D	FIR	; ELSE jump to address	
	0830	22	• • • • • • • • • • • • • • • • • • • •	STRET:	RET			
							; Return	

Same				; ;	ASM -	ANSI set mode sequen	ce
ENTRY	. 3			,			
					FNTRY	none	
USES all USES all							
11							
13 083D 31 D1	11						
15 0841 75 FO	13 083D	31 D		ASM:		ASMR	; IF parameter THEN
17 0847 31 F7 ACALL XSTAB search table 18 0849 40 04 JC ASMR IF found THEN 19 0848 11 3A ACALL STCALL i jump to revisine 20 084D 30 EE SJMP ASM and keep soins 21 084F 22 ASMR: RET	15 0841	75 F.	005 850			DPTR, #ASMT	; (DPTR) = "ESC [Pn h" table
19 084B 11 3A	17 0847	31 F				ASMR	; search table ; IF found THEN
21	19 084B	113	A				
23 24 25 26 27.	21		-	ASMR:	RET		
31 32						0 h (C) 7	
34 0850 01							
36 37 0853 02 DB 2 ; ESC [2 h 38 0854 0FB9 DW DKI ; Disable keyboard input 39 40 0856 04 DB 4 ; ESC [4 h 41 0857 0854 DW EICM ; Enter insert character mode 42 43 0859 06 DB 6 ; ESC [6 h 44 085A 0874 DW EERM ; Enter erasure mode 45 46 085C 14 DB 20 ; ESC [20 h 47 085D 0862 DW EACR ; Enable auto CR on LF	28 29 30 31 32	0850		ASMT	*ASMT*	contains.valid.mans	
38 0854 0FB9	28 29 30 31 32 33 34 0850	01		i; ; ,	*AŞMT* EQU DB	s i	meters for "ESC [Pn h" sequence. ; ESC [1 h
40 0856 04 DB 4 ; ESC L 4 h 41 0857 0854 DW EICM ; Enter insert sharacter mode 42 43 0859 06 DB 6 ; ESC L 6 h 44 085A 0874 DW EERM ; Enter erasure mode 45 DB 20 ; ESC L 20 h 47 085D 0862 DW EACR ; Enable auto CR on LF	28 29 30 31 32 33 34 0850 35 0851	01		i, i,	*AŞMT* EQU DB DW	s 1 EGATM	meters for "ESC [Pn h" sequence. ; ESC [1 h ; Enter quarded area transfer
42 43 0859 06 DB 6 ; ESC [6 h 44 085A 0874 DW EERM ; Enter erasure mode 45 DB 20 ; ESC [20 h 47 085D 0862 DW EACR ; Enable auto CR on LF 48	28 29 30 31 32 33 34 0850 35 0851 36 37 0853	01 .086E		asmt	*AŞMT* EQU DB DW DW	s 1 EGATM	meters for "ESC [Pn h" sequence. ; ESC [1 h ; Enter guarded area transfer. ; ESC [2 h
44 085A 0B74 DW EERM ; Enter erasure mode 45 46 085C 14 DB 20 ; ESC [20 h 47 085D 0B62 DW EACR ; Enable auto CR on LF 48	28 29 30 31 32 33 34 0850 35 0851 36 37 0853 38 0854 39	01 .084E 02 .0FB9		i; ; ; ASMT	*ASMT* EQU DB DW DB DW	s contains valid pars s 1 EGATM 2 DKI	meters for "ESC [Pn h" sequence. ; ESC [1 h ; Enter quanded area transfer ; ESC [2 h ; Disable keyboard input ; ESC [4 h
46 085C 14 DB 20 ; ESC L 20 h 47 085D 0862 DW EACR ; Enable auto CR on LF 48	28 29 30 31 32 33 34 0850 35 0851 36 37 0853 38 0854 39 40 0856 41 0857	01 .086E		ASMT	*ASMT* EQU DB DW DB DW DB DW	s 1 EGATM DKI 4 EICM	meters for "ESC [Pn h" sequence. ; ESC [1 h ; Enter guarded area transfer ; ESC [2 h ; Disable keyboard input ; ESC [4 h ; Enter insert character mode.
48	28 29 30 31 32 33 34 0850 35 0851 36 37 0853 38 0854 39 40 0856 41 0857 42 43 0859	01 .086E		ASMT	*ASMT* EQU DB DW DB DW DB DW DB DW	s 1 EGATM 2 DKI 4 EICM	meters for "ESC [Pn h" sequence. ; ESC [1 h ; Enter guarded area transfer ; ESC [2 h ; Disable keyboard input ; ESC [4 h ; Enter insert character mode ; ESC [6 h
49 0005 ASMTL EQU (\$-ASMT)/3 tTable length	28 29 30 31 32 33 34 0850 35 0851 36 37 0853 38 0854 39 40 0856 41 0857 42 43 0859 44 085A 45	01 086E		asmt	#AŞMT# EQU DB DW DB DW DB DW DB DW	s 1 EGATM 2 DKI 4 EICM 6 EERM	meters for "ESC [Pn h" sequence, ; ESC [1 h ; Enter guarded area transfer ; ESC [2 h ; Disable keyboard input ; ESC [4 h ; Enter insert character mode ; ESC [6 h ; Enter erasure mode
	28 29 30 31 32 33 34 0850 35 0851 36 37 0853 38 0854 39 40 0856 41 0857 42 43 0859 44 085A 45 46 085C 47 085D 48	01 086E		ASMT	*ASMT* EQU DB DW DB DW DB DW DB DW DB DW DB DW	s s s s s s s s s s s s s s s s s s s	meters for "ESC [Pn h" sequence, ; ESC [1 h ; Enter guarded area transfer ; ESC [2 h ; Disable keyboard input ; ESC [4 h ; Enter insert character mode ; ESC [6 h ; Enter erasure mode ; ESC [20 h ; Enable auto CR on LF
	28 29 30 31 32 33 34 0850 35 0851 36 37 0853 38 0854 39 40 0856 41 0857 42 43 0859 44 085A 45 46 085C 47 085D 48	01 086E		ASMTL	#ASMT# EQU DB DW DB DW DB DW DB DW EQU	\$ 1 EGATM 2 DKI 4 EIGM 6 EERM 20 EACR	### ##################################
	28 29 30 31 32 33 34 0850 35 0851 36 37 0853 38 0854 39 40 0856 41 0857 42 43 0859 44 085A 45 46 085C 47 085D 48	01 086E		ASMTL	#ASMT# EQU DB DW EQU	\$ 1 EGATM 2 DKI 4 EICM 6 EERM 20 EACR (\$-ASMT)/3	### ##################################

					ANSI reset mode see	uence.
4		• • • • • • • • • • • • • • • • • • • •	; 	*ARM*	reset the specified	mode(s).
			; -			. 20 20 4 4 5 5 4 5 5 4
۵	• • • • • • • • • • • • • • • • • • • •	•••••		ENTRY	none	
9			; •	EXIT	none	
9 10			• • • • • • • • • • • • • • • • • • •	USES	all	·····
	085F 31 0861 40	D1 ÉC	ARM:	ACALL	GNP	; REPEAT set parameter
				JĊ	ASMR	; IF no parameter THEN return
	0863 75 0866 90	F0	.05	MOV	B, #ARMTL	; (B) = table length
		0871		MOV	DPTR, #ARMT	; (DPTR) = "ESC [Pn]" table
	0869 31	F7 E2		ACALL	XSTAB	; search table
	086B 40	E2		JC	ASMR	; IF not found THEN return
	086D 11 086F 80	3A		ACALL	STCALL	; call routine
20	086F 80	EE		SJMP	ARM	; UNTIL forever
21						2 marriage two ways
22						
23 2 4						
25			5 5	ARMT -	ANSI reset mode se	quence table
26			,			· · · · · · · · · · · · · · · · · · ·
27 28			;	*ARMT*	contains valid para	ameters for "ESC [Pn]" sequence
28						**************************************
29						
29 30		i	ARMT	EQU	\$	
31		î	ARMT		**************************************	
31 32	087101	-	ARMT	EQU	*	: FSC [1]
31 32 33		-	ÄŔMŤ	DB	_i	; ESC [1]
31 32	087101	-	ARMŤ			; ESC [1] ; Exit auarded area transfer
31 32 33 34	0871 01 0872 087 0874 02	1	ARMŤ	DB DW	i XGATM	; Exit guarded area transfer
31 32 33 34 34 35	0871 01 0872 087	1	ARMT	DB DW	i XGATM	; Exit guarded area transfer
31 32 33 34 35	0871 01 0872 087 0874 02 0875 0FB	1	ÄŘMŤ	DB	_i	
31 32 33 34 35 36 37	0871 01 0872 087 0874 02 0875 0FB	1	ÄŘMŤ	DB DW DB	i XGATM 2 EKI	; Exit suarded area transfer ; ESC [2] ; Enable keyboard input
31 32 33 34 35 36 37	0871 01 0872 087 0874 02 0875 0FB	3	ARMT	DB DW DB DW	i XGATM 2 EKI	; Exit guarded area transfer ; ESC [2] ; Enable keyboard input ; ESC [4]
31 32 33 34 35 36 37	0871 01 0872 087 0874 02 0875 0FB	3	ARMT	DB DW DB	i XGATM 2 EKI	; Exit guarded area transfer ; ESC [2] ; Enable keyboard input
31 32 33 34 35 36 37 38 39 40	0871 01 0872 087 0874 02 0875 068 0877 04 0878 085	3	ARMT	DB DW DB DB DB	i XGATM 2 EKI 4 XICM	; Exit guarded area transfer ; ESC [2] ; Enable keyboard input ; ESC [4] ; Exit insert character mode
31 32 33 34 35 36 37 38 39 40	0871 01 0872 0B7 0874 02 0875 0FB 0877 04 0878 0B5	3	ARMT	DB DW DB DB DB	i XGATM 2 EKI 4 XICM	; Exit guarded area transfer ; ESC [2] ; Enable keyboard input ; ESC [4] ; Exit insert character mode
31 32 33 34 35 36 37 38 39 40 41	0871 01 0872 087 0874 02 0875 068 0877 04 0878 085	3	ARMT	DB DW DB DW	i XGATM 2 EKI	; Exit suarded area transfer ; ESC [2] ; Enable keyboard input ; ESC [4]
31 32 33 34 35 36 37 38 39 40 41 42 43	0871 01 0872 087 0874 02 0875 068 0877 04 0878 085 0878 06	3	ARMT	DB DW DB DW DB DW DB	i XGATM 2 EKI 4 XICM 6 XERM	; Exit guarded area transfer ; ESC [2] ; Enable keyboard input ; ESC [4] ; Exit insert character mode ; ESC [6] ; Exit erasure mode
31 32 33 34 35 36 37 38 39 40 41 42 43	0871 01 0872 087 0874 02 0875 068 0877 04 0878 085 087A 06 087B 087	1 3 8	ARMT	DB DW DB DW DB DW DB DW DB DW	1 XGATM 2 EKI 4 XICM 6 XERM	; Exit guarded area transfer ; ESC [2] ; Enable keyboard input ; ESC [4] ; Exit insert character mode ; ESC [6] ; Exit erasure mode ; ESC [20]
31 32 33 34 35 36 37 38 39 40 41 42 43 44	0871 01 0872 087 0874 02 0875 068 0877 04 0878 085 0878 06	1 3 8	ARMT	DB DW DB DW DB DW DB	i XGATM 2 EKI 4 XICM 6 XERM	; Exit guarded area transfer ; ESC [2] ; Enable keyboard input ; ESC [4] ; Exit insert character mode ; ESC [6] ; Exit erasure mode ; ESC [20]
31 32 33 34 35 36 37 38 39 40 41 42 43	0871 01 0872 087 0874 02 0875 068 0877 04 0878 085 087A 06 087B 087	1 3 8 7	ARMT ARMTL	DB DW DB DW DB DW DB DW DB DW	1 XGATM 2 EKI 4 XICM 6 XERM	; Exit suarded area transfer ; ESC [2] ; Enable keyboard input ; ESC [4] ; Exit insert character mode ; ESC [6] ; Exit erasure mode

			A1M	ANSI mode #1	
1				••••	
			A1M	inputs the parameter SC [? Ps F sequence	string and final character for
4 5		;	tne E	SC [? FS F Sequent	
			ENTRY	none	
9		,	EXIT	none	
10		;	USES	all	
.11					
.13	4.00	A1M:	CLR	PSDF	; Clear *PSD* flas
14 0880 C2 15 0882 11	18 2A	HIII.	ACALL	SETDISP	; Set dispatch and return
16			ACALL	PSD	; Get parameter string
17 0884 31 18 0886 50	71 67		JNC	ASMR	; IF not done THEN return
19 0888 11	2F		ACALL.	SETNORM	; ELSE restore dispatch ; (B) = table length
	F0 02 0892		MOV MOV	B, #EQMTL DPTR, #EQMT	; (DPTR) = ESC [? table
21 088D 90 22 0890 01	36		AJMP	STJMP	; Jump to any routine
.23					
24 25					
26			FOMT -	escape [? Pn Ps t	able
27 28					
.29			*EQMT*	contains the fourt	h and/or final character
- 30 31					
32 08	92	EQMT	EQU	\$	
33 34 0892 68			DB		; ESC [? Pn h
35 0893 08			DW	A1SM	; ANSI set mode #1
- 36 - 37. 0895 60			DB DW	(1/	; ESC [? Pn]
38 0896 08			DW	A1RM	; ANSI reset mode
39 40 00	 02	EQMTL	EQU	(\$-EQMT)/3	; Table length

····· <u>.</u> · · · · · · · · ·				A2M	ANSI mode #2	
3			;			
4 =				the E	ESC [> Pn Ps seque	n.string.and.final.character.for
6			· · · · · · · · · · · · · · · · · · ·			
7	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		ENTRY	none	
<u>?</u>			,			
10 11			; ;	HSES	~ 1 1	
12 13					9417	
14 0898	C2	18	A2M:	CLR	PSDF	; Clear *PSD* flas
15_089A. 16	1.1	2A	.,	AÇALL.	SETDISP	
17 0890	31 50	71		ACALL	PSD	
18 089E 19 08A0	50 11	22 2F		JNC	A1SR	; IF not done THEN return
20 08 A 2	75	FO 02		AÇALL MOV	SETNORM B, #EGTTL	
2108A5. 22.08A8	99 01	98AA 36		MQY	DPTRx.#EGTT	
23				AJMP	SIJMP	; Jump to any routine
2 4 25						·····
26						
27 28				EGTT -	escape [> Pn Ps t	able
27 28						able
27 28 29 30			5.5 5 5			
27 28 29 30 31 31	OSAA		EGTT			able
27 28 29 30 31 32 33 34 08AA	68		;	*EGTT*	contains the fourt	able h and/or final character.
27 28 29 30 31 32 33 34 08AA 35 08AB			;	*EGTT*	contains the fourt	able h and/or final character. s ESC [> Pn h
27 28 29 30 31 32 33 34 08AA 35 08AB 36	68 08FF 60		;	*EGTT* EQU DB	contains the fourt \$ 'h' A2SM	able h and/or final character. ; ESC [> Pn h ; ANSI set mode #1
27 28 29 30 31 32 33 34 08AA 35 08AB 36 37 08AD 38 08AE	68 08FF		;	*EGTT* EQU DB DW	contains the fourt \$ 'h'	able h and/or final character. s ESC [> Pn h
27 28 29 30 31 32 33 34 08AA 35 08AB 36	68 08FF 60 0944		EGTT	*EGTT* EQU DB DW DB DW	contains the fourt \$ 'h' A2\$M 11' A2RM	able h and/or final character. ; ESC [> Pn h ; ANSI set mode #1 ; ANSI reset mode
27 28 29 30 31 32 33 34 08AA 35 08AB 36 37 08AD 38 08AE	68 08FF 60		EGTTL	*EGTT* EQU DB DW DB DW EQU	contains the fourt \$ 'h' A2\$M	able h.and/or final character. ; ESC [> Pn h ; ANSI set mode #1
27 28 29 30 31 32 33 34 08AA 35 08AB 36 37 08AD 38 08AE	68 08FF 60 0944		EGTTL	*EGTT* EQU DB DW DB DW EQU	contains the fourt \$ 'h'A2SM A2RM (\$-EGTT)/3	able h and/or final character. ; ESC [> Pn h ; ANSI set mode #1 ; ANSI reset mode
27 28 29 30 31 32 33 34 08AA 35 08AB 36 37 08AD 38 08AE	68 08FF 60 0944		EGTTL	*EGTT* EQU DB DW DB DW EQU	contains the fourt \$ 'h'A2SM A2RM (\$-EGTT)/3	able h and/or final character. ; ESC [> Pn h ; ANSI set mode #1 ; ANSI reset mode
27 28 29 30 31 32 33 34 08AA 35 08AB 36 37 08AD 38 08AE	68 08FF 60 0944		EGTTL	*EGTT* EQU DB DW DB DW EQU	contains the fourt \$ 'h'A2SM A2RM (\$-EGTT)/3	able h and/or final character. ; ESC [> Pn h ; ANSI set mode #1 ; ANSI reset mode
27 28 29 30 31 32 33 34 08AA 35 08AB 36 37 08AD 38 08AE	68 08FF 60 0944		EGTTL	*EGTT* EQU DB DW DB DW EQU	contains the fourt \$ 'h'A2SM A2RM (\$-EGTT)/3	able h and/or final character. ; ESC [> Pn h ; ANSI set mode #1 ; ANSI reset mode
27 28 29 30 31 32 33 34 08AA 35 08AB 36 37 08AD 38 08AE	68 08FF 60 0944		EGTTL	*EGTT* EQU DB DW DB DW EQU	contains the fourt \$ 'h'A2SM A2RM (\$-EGTT)/3	able h and/or final character. ; ESC [> Pn h ; ANSI set mode #1 ; ANSI reset mode
27 28 29 30 31 32 33 34 08AA 35 08AB 36 37 08AD 38 08AE 39 40	68 08FF 6C 0944 0002		EGTTL	*EGTT* EQU DB DW DB DW EQU	contains the fourt \$ 'h' .A2SM A2RM (\$-EGTT)/3	able h and/or final character. ; ESC [> Pn h
27 28 29 30 31 31 32 33 34 08AA 35 08AB 36 37 08AD 38 08AE 39 40	68 08FF 6C 0944 0002		EGTT EGTTL	#EGTT# EQU DB DW DB DW EQU	contains the fourt \$ 'h' A2SM -'1' A2RM (\$-EGTT)/3	able h and/or final character. ; ESC [> Pn h ; ANSI set mode #1 ; ESC [> Pn l ; ANSI reset mode ; Table length
27 28 29 30 31 32 33 34 08AA 35 08AB 36 37 08AD 38 08AE 39 40	68 08FF 6C 0944 0002		EGTT EGTTL	#EGTT# EQU DB DW DB DW EQU	contains the fourt \$ 'h' A2SM -'1' A2RM (\$-EGTT)/3	able h and/or final character. ; ESC [> Pn h ; ANSI set mode #1 ; ANSI reset mode
27 28 29 30 31 32 33 34 08AA 35 08AB 36 37 08AD 38 08AE 39	68 08FF 6C 0944 0002		EGTTL	#EGTT# EQU DB DW DB DW EQU	contains the fourt \$ 'h' .A2SM A2RM (\$-EGTT)/3	able h and/or final character. ; ESC [> Pn h ; ANSI set mode #1 ; ESC [> Pn l ; ANSI reset mode ; Table length

SAISHW set the specified mode(s).	1			AISM -	ANSI mode #1 set mo	de sequence
STATE STAT	2		5			
ENTRY none	.3 .4			*#1201*	set the specified in	044574
	5			ENTRY	none	
	7					
1	8 9		;			
3 3 3 3 3 3 3 4 3 3	10		,	USES	A, B, DPTR	
4 0802 40 0E	12			00011	CND	: Got nevt (or first) parameter
5 0884 75 FO 03 MOV B, #AISTL 1 (B) = table length 6 0887 90 0863 MOV DPTR #AIST 1 DPTR = ESC [? Ph h table 7 0883 1, F7 ACALL XSTAB 1, Search table 8 0886 40 04 JC AISR 1 F found THEN 9 0886 1 11 3A ACALL STCALL 1 Jume to routine 0 0800 8E SMP AISM 1 and keep goins 1 1 3A ACALL STCALL 1 Jume to routine 0 0800 8E SMP AISM 1 and keep goins 1 1 20862 22 AISR: RET 1 1 AIST - ANSI mode #1 set mode table 1 1 AIST - ANSI mode #1 set mode table 1 1 AIST - ANSI mode #1 set mode table 1 1 AIST - ANSI MODE #1 SEC [? Ph. b. sequence. 1 AIST 2 O863 AIST EOU \$ 1 ESC [? 2 h		D1	Albh:			; IF parameter THEN
\$ 6 0887 90 08C3 MOV DPTR #AIST ! UPIN = ESC ! ?P N table 7 088A 31 F7 ACALL XSTAB ! Search table 8 088C 40 04 JC AISR ! IF found THEN 9 088E 11 3A ACALL STOAL ! June to routine 9 088C 51 3A ACALL STOAL ! June to routine 9 088C 51 3A ACALL STOAL ! June to routine 9 088C 51 3A ACALL STOAL ! June to routine 9 088C 51 3A ACALL STOAL ! June to routine 9 088C 51 3A ACALL STOAL ! June to routine 9 088C 51 3A ACALL STOAL ! June to routine 9 088C 51 3A ACALL STOAL ! June to routine 9 088C 51 3A ACALL STOAL ! June to routine 9 088C 51 3A ACALL STOAL ! June to routine 9 088C 51 3A ACALL STOAL ! June to routine 9 088C 51 3A ACALL STOAL ! June to routine 9 088C 51 3A ACALL STOAL ! June to routine 9 088C 51 3A ACALL STOAL ! June to routine 9 088C 51 3A ACALL STOAL ! June to routine 9 088C 51 3A ACALL STOAL ! June 10 088C 51 3A ACALL STOAL * June 10 088C 51 3A ACAL STOAL * June 10 088C 51 3A ACALL STOAL * June 10 088C 51 3A ACAL STOAL * June 10 088C 51 3A ACALL STOAL * June 10 088C 51 3A ACALL STOAL * June 10 088C 51 3A ACALL STOAL * June 10 088C 51 3A A						
7. OBBA 31 F7 ACALL XSTAB SEARCH TABLE. 8 OBBC 40 04 JC AISR I IF found THEN OF 9. OBBE 11 3A ACALL STCALL I JUMP to routine O OBCO 80 EE SUMP AISM I and keer soins 2 OBCC 22 AISR RET 3	6 08B7 90	0803				; (DPTR) = ESC [? Pn h table
8 GBBC 40 04 JC A1SR ; IF found them. 9 OBBC 11 3A ACALL STCALL 1 Jump to routine 0 OBCO 80 EE SUMP A1SM 1 and keep soins 1	7 08BA 31	F7				
0 08C0 80 EE SJMP AISM and keep soins 2 08C2 22 AISR: RET 34		04				
2 08C2 22 AISR: RET 3		3A				
2 08C2 22 AISR: RET 3	01	EE			AISM	; and keep actua
4	2 0802 22		A1SR:	RET		
5				• • • • • • • • • • • • • • • • • • • •		
S	5					
#AISMT* contains valid Parameters for ESC I. 2 Pn.h. sequence. #AISMT* contains valid Parameters for ESC I. 2 Pn.h. sequence. #AISMT* contains valid Parameters for ESC I. 2 Pn.h. sequence. #AISMT* contains valid Parameters for ESC I. 2 Pn.h. sequence. #AISMT* contains valid Parameters for ESC I. 2 Pn.h. sequence. #AISMT* contains valid Parameters for ESC I. 2 Pn.h. sequence. #BIST I						
0				A1ST	ANSI mode #1 set mo	de table
1	7 8		5. ‡			
33 34 OSC3 O2 DB 2 ; ESC [? 2 h 35 OSC4 OBO7. DW EZM i.Enter ZDS mode 36 37 OSC6 O3 DB 3 ; ESC [? 3 h 38 OSC7 OBAC DW CLRS ; Enter 132 column mode 39 30 OSC9 O7 DB 7 ; ESC [? 7 h 31 OSCA OAFD DW WEOL i.Wrap.at.cod.cf.lios 32 33 OOO3 A1STL EQU (\$-A1ST)/3 i.Table.length			5.5			
DB 2 FEST 7 3 FEST	7. 8 9 0 1		5.5	*A1SMT	*.contains.valid.par	
2007 2007	77. 28. 29. 30. 31.	3	ii i i	*A1SMT EQU	* contains valid par	ameters for ESC [? Pn h sequence.
DB 3 ESC L 7 3 5 580 087 086C DW CLRS Enter 132 column mode 38 0867 086C DW CLRS Enter 132 column mode 39 16 0869 07 DB 7 ESC [? 7 h 10 086A QAFD DW WEQL I Wrap at end of line 12 12 12 13 14 15 15 15 15 15 15 15	7. 28 29 30 31 32 08C 33 34 08C3 02	_	,; , , , A1ST	*A1SMT EQU DB	* contains valid par \$	ameters for ESC [? Pn h sequence. ; ESC [? 2 h
SB 08C7 OBAC DW CLRS Enter 132 column mode SP SP SP SP SP SP SP S	27 28 29 30 31 32 32 33 34 35 35 36 37 38 38 38 38 38 38 38 38 38 38 38 38 38	_	;; ; ; A1ST	*A1SMT EQU DB	* contains valid par \$	ameters for ESC [? Pn h sequence. ; ESC [? 2 h
DB	7. 8. 9. 10. 11. 12. 08C3 13. 14. 08C3 02 15. 08C4 0801	7	;; ; ; A1ST	*A1SMT EQU DB DW	* contains valid Par \$ 2 EZM	ameters for ESC [? Pn h sequence. ; ESC [? 2 h ; Enter ZDS mode. ; ESC [? 3 h
OBCA OAFD DW WEOL	7.89.99.00.00.00.00.00.00.00.00.00.00.00.00	7	ii ; ; AIST	*A1SMT EQU DB DW DB	* contains valid Par \$ 2 EZM	ameters for ESC [? Pn h sequence. ; ESC [? 2 h ; Enter ZDS mode. ; ESC [? 3 h
13 0003 AISTL EQU (\$-AIST)/3 ; lable length	7. 28 29 30 32 32 34 08C3 02 35 08C4 080 36 37 08C4 080 38 39 08C7 08A	7	AIST	*A1SMT EQU DB DW DB DW	* contains valid par \$ 2 EZM CLRS	ameters for ESC [? Pn h sequence. ; ESC [? 2 h ; Enter ZDS mode. ; ESC [? 3 h ; Enter 132 column mode
	7	7	,; , , A1ST	*A1SMT EQU DB DW DB DW	* contains valid par \$ 2 EZM CLRS	ameters for ESC [? Pn h sequence. ; ESC [? 2 h ; Enter ZDS mode. ; ESC [? 3 h ; Enter 132 column mode ; ESC [? 7 h
	27. 28. 29. 30. 31. 32. 33. 34. 08C3. 35. 08C4. 385. 08C4. 386. 37. 08C6. 38. 08C7. 388. 399. 399. 399. 399. 399. 399. 399	7		*A1SMT EQU DB DW DB DW DB	* contains valid par \$ 2 EZM 3 CLRS 7 WEOL	ameters for ESC E ? Pn h sequence. ; ESC E ? 2 h ; Enter ZDS mode. ; ESC E ? 3 h ; Enter 132 column mode ; ESC E ? 7 h ; Wrap at end of line.
	27. 28 29 29 30 31 32 32 33 34 08C3 02 35 08C4 0803 36 37 08C6 03 38 08C7 08A6 39 08C9 07 41 08CA 0AF1	7		*A1SMT EQU DB DW DB DW DB	* contains valid par \$ 2 EZM 3 CLRS 7 WEOL	ameters for ESC E ? Pn h sequence. ; ESC E ? 2 h ; Enter ZDS mode. ; ESC E ? 3 h ; Enter 132 column mode ; ESC E ? 7 h ; Wrap at end of line.
	27. 28 29. 30 31. 32 32 33 34 08C3 02 35 08C4 08D3 36 37 08C6 03 38 08C7 08A6 37 08C6 07 41 08C9 07	7		*A1SMT EQU DB DW DB DW DB	* contains valid par \$ 2 EZM 3 CLRS 7 WEOL	ameters for ESC E ? Pn h sequence. ; ESC E ? 2 h ; Enter ZDS mode. ; ESC E ? 3 h ; Enter 132 column mode ; ESC E ? 7 h ; Wrap at end of line.
	27. 28. 29. 29. 30. 31. 32. 33. 34. 08C3. 02. 35. 08C4. 080. 37. 08C6. 03. 38. 08C7. 08A6. 37. 08C9. 07.	7		*A1SMT EQU DB DW DB DW DB	* contains valid par \$ 2 EZM 3 CLRS 7 WEOL	ameters for ESC E ? Pn h sequence. ; ESC E ? 2 h ; Enter ZDS mode. ; ESC E ? 3 h ; Enter 132 column mode ; ESC E ? 7 h ; Wrap at end of line.
	27. 28. 29. 30. 31. 32. 33. 34. 08C3. 35. 08C4. 385. 08C4. 386. 37. 08C6. 38. 08C7. 388. 399. 399. 399. 399. 399. 399. 399	7		*A1SMT EQU DB DW DB DW DB	* contains valid par \$ 2 EZM 3 CLRS 7 WEOL	ameters for ESC E ? Pn h sequence. ; ESC E ? 2 h ; Enter ZDS mode. ; ESC E ? 3 h ; Enter 132 column mode ; ESC E ? 7 h ; Wrap at end of line.
	27, 28 29, 29, 30, 31, 32, 08C3 33, 35, 08C4 980, 35, 08C4 980, 38, 08C7 984, 39, 40, 08C9 07, 41, 08CA 942, 43, 000	7 D	A1STL	#A1SMT EQU DB DW DB DW DB DW	* contains valid Par \$ 2 EZM 3 CLRS 7 WEOL (\$-AIST)/3	ameters for ESC L ? Pn h sequence. ; ESC L ? 2 h ; Enter ZDS mode ; ESC L ? 3 h ; Enter 132 column mode ; ESC L ? 7 h ; Wrap at end of line ; Table length
	27. 28 29. 29. 30. 31. 32. 33. 34. 08C3. 35. 08C4. 0803. 36. 37. 08C4. 03. 38. 08C7. 08A8. 39. 40. 08C9. 41. 08CA. 0AF. 42. 43. 0003	7 C D 3	A1STL	#A1SMT EQU DB DW DB DW DB DW	* contains valid Par \$ 2 EZM 3 CLRS 7 WEOL (\$-AIST)/3	ameters for ESC E ? Pn h sequence. ; ESC E ? 2 h i Enter ZDS mode ; ESC E ? 3 h ; Enter 132 column mode ; ESC E ? 7 h i Wrae at end of line ; Table length
	27, 28 27, 28 29, 29, 30, 31, 32 08C3 33, 34, 08C3 02, 35, 08C4 08C3 36, 37, 08C6 03, 38, 08C7 08A6, 39, 40, 08C9 07, 41, 08CA 0AF1, 42, 43 0000	7	A1STL	*A1SMT EQU DB DW DB DW DB DW EQU	* contains valid par \$ 2 EZM 3 CLRS 7 WEOL (\$-AIST)/3	ameters for ESC E ? Pn h sequence. ; ESC E ? 2 h ; Enter ZDS mode. ; ESC E ? 3 h ; Enter 132 column mode ; ESC E ? 7 h ; Wrap at end of line. ; Table length

Z29ROM; MICRO *** Z-29 COMP	BENCH 805	1 CROSS ASSEMBLE	ER (V1)-218 24-	FEB-83 15	5:30:06 PAGE 76	
1		errole in this leading W	:;		- ANSI mode #1 nee	et mode sequence
2 3			,			
				*A1RM	*.resets the speci	fied mode(s).
······6 7	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		ENTRY	none	
8 ,9		••••••		EXIT	none	
10 11	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	····	USES	A, B, DPTR	
12 13 08CC		****		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
14 08CE	31 40	<u>D1</u> F2	A1RM:	ACALL. JC	GNP A1SR	; REPEAT set parameter
15 08D0	75 90			MOV	B, #A1RTL	; IF no parameter THEN exit
16 08D3		OSDE	••••••	····MŏV····	DPTR, #AIRT	; (B) = table length ; (DPTR) = ESC [? Pn table
17 08D6	31 40	<u>F7</u> E8		ACALL	XSTAB	Search table
18 0808			••••••	JĊ	AISR	; IF found THEN
19 08DA	1 1	3A EE		ACALL	STCALL	jump to routine
20 08DC	80	EE		SUMP	A1RM	; UNTIL forever
21 22 23	• • • • • • • • • • • • • • • • • • • •					
23 24 25 26	• • • • • • • • • • • • • • • • • • • •	•••••	•••••••••••••••••••••••••••••••••••••••	Δ1FT -	- ANSI mode #1 res	
26			***************************************	: : * : : :		er mode (4b)6
27 28				*A1RMT	* contains valid	parameters for ESC [? Pn] sequence.
29 30						
31	OSDE		AIRT	EģÜ	\$	
32 08DE 33 08DF	02 0B07			DB	···2	; ESC [? 2]
34		• • • • • • • • • • • • • • • • • • • •		<u>D</u> W	EZM	; Enter VT52 mode
35 08E1	03			DB DW	3 CLRS	; ESC [? 3]
36 08É2 37	ÓBÁC				CLRS	; Enter 80 column mode
38 08É4	····òż······		• • • • • • • • • • • • • • • • • • • •	ĎĖ	····›	; ESC [? 7]
39 08E5 40	OBOO	• • • • • • • • • • • • • • • • • • • •		DW	DEOL	Discard Past end of line
41	0003		A1RTL	EQU	(\$-A1RT)/3	; Table length
		• • • • • • • • • • • • • • • • • • • •	•••••		,	
•••••						
	• • • • • • • • • • • • • • • • • • • •				•••••	
					•••••	
	• • • • • • • • • • • • • • • • • • • •		••••••			
		•••••				
	•••••	••••••				

Z29R0	M; MICROÉ	ENCH 805	1 CROSS A	ASSÉMBLÉR MWARE VERS	(V1)-218 24 ION 1.03 ***	4-FEB-83 15:	30:06 PAGE 77		
		!!EN!!E!!!			;;		et mode sequence		
	1 2 3 4				; ;	*SMS* s	ets the mode specified	by the last character in th	e
	.5 					ENTRY	none		
•••••	. 7 9				; ;	EXIT	none		
	9 10 11 12				; ;	USES	all		
	12 13 14 08E7	11	2A		SMS:	······ACALĹ	SETDISP	; Set dispatch and return	 I
•••••	15 16 08E9		2F			ACALL	SETNORM	; Restore dispatch addres	s
	17 08EB 18 08EC	Ç3 94	зі			CLR SUBB	C A, #/1/	; Subtract offset	
************	19 08EE 20 08F0	40 B4	OE 09	oo		JC CUNE JNC	SMSRET A, #(/9/-/0/), SMS1 SMSRET	; Check for lower bounds ; Check for upper bounds	
	21 08F3 22 08F5 23 08F4	50 04 75	09 F0	09	SMS1:	INC MOV	A B, #SMSTL	; Bump up into range ; (B) = table length	
	23 08F6 24 08F9 25 08FC	75 90 01	F0 0911 36		• • • • • • • • • • • • • • • • • • • •	MOV AJMP	DPTR, #SMST STJMP	; (DPTR) = ESC x Pn table ; Jump to any routine	
	25 08FC 26 27 08FE	22			SMSRET:	RET			
								······································	
	• • • • • • • • • • • • • • • • • • • •								

)))

)

•

1					
·······	• • • • • • • • • • • • • • • • • • • •		A2SM.:	- ANSI mode #2 set m	node sequence
		,	*A2SM	* set the specified	mode(s)
4		*		1.898.808.80841.499.	mode (57a
6 7		;	ENTRY	none	
8			EXIT		
		,		none	
10		*	USES	A, B, DPTR	
11					
12 13.08FF	31 D1	Астем.	00011		
14 0901	40 FB	A2SM:	ACALL. JC	GNP SMSRET	
15.0903			MOV	B, #SMSTL	; IF parameter THEN
16 0906	90 09:		MOV	DPTR, #SMST	; (B) = table length
17,0909			ACALL.	XSTAB	; (DPTR) = ESC [> Pn h table
18 090B	31 F7		JC	SMSRET	; Search table ; IF found THEN
19020D	3A.		ACALL	STCALL	
20 090F	80 EE		JMP	A2SM	jump to routine and keep going
21		••••••			, gud yeek adilla
22					
23					
24 25					
····· <i>+</i> ×······ 26		•••••••	SMST	set mode sequence.	table
27		<u>.</u>			
+ / 28		••••••	*\$M\$J.*	t contains valid par	ameters for ESC × Pn sequence
					* * * * * * * * * * * * * * * * * * *
		,	and E	SC [> Pn h sequen	ce
22 30			and E	SC [> Pn h sequen	ce
30	 0911	,	and E	SC [> Pn h sequen	Ce
	0911	, SMST	and E	SC [> Pn h sequen	Ce
30 31 32 33 0911	01	, SMST	EQU	SC L > Pn h sequen	ce
30 31 32 33 0911 34 0912		, SMST	EQU	SC L > Pn h sequen	ce
30 31 32 33 0911 34 0912 35	01 0B0B	smst	EQU DB DW	\$	ce
30 31 32 33 0911 34 0912 35 36 0914	01 0808 02	, SMST	EQU	\$	ce ; ESC × 1 ESC [> 1 h ; Enable 25th line
30 31 32 33 0911 34 0912 35 36 0914 37 0915	01 0B0B	, SMST	EQU DB DW	\$	<pre></pre>
30 31 32 33 0911 34 0912 35 36 0914 37 0915	01 0B0B 02 0B2D	, SMST	EQU DB DW DB DW	\$ 1 E25L 2 NKC	ce ; ESC × 1 ESC [> 1 h ; Enable 25th line
30 31 32 33 0911 34 0912 35 36 0914 37 0915 38 39 0917	01 0808 02 0820 03	smst	EQU DB DW DB DW	\$ 1 E25L 2 NKC	ce ; ESC × 1 ESC [> 1 h ; Enable 25th line ; ESC × 2 ESC [> 2 h ; No keyboard click
30 31 32 33 0911 34 0912 35 36 0914 37 0915 38 39 0917 40 0918	01 0B0B 02 0B2D	smst	EQU DB DW	\$	<pre></pre>
30 31 32 33 0911 34 0912 35 36 0914 37 0915 38 39 0917 40 0918 41	01 0808 02 0820 03 083C	smst	EQU DB DW DB DW DB	\$ 1 E25L 2 NKC 3 EHSM	ce ; ESC × 1 ESC [> 1 h ; Enable 25th line ; ESC × 2 ESC [> 2 h ; No keyboard click ; ESC × 3 ESC [> 3 h ; Enter hold screen mode
30 31 32 33 0911 34 0912 35 36 0914 37 0915 38 39 0917 40 0918 41 42 091A	01 0B0B 02 0B2D 03 0B3C	smst	EQU DB DW DB DW DB DW	\$	ce ; ESC × 1 ESC [> 1 h ; Enable 25th line ; ESC × 2 ESC [> 2 h ; No keyboard click ; ESC × 3 ESC [> 3 h ; Enter hold screen mode ; ESC × 4 ESC [> 4 h
30 31 32 33 0911 34 0912 35 36 0914 37 0915 38 39 0917 40 0918 41 42 091A 43 091B	01 0808 02 0820 03 083C	SMST	EQU DB DW DB DW DB	\$ 1 E25L 2 NKC 3 EHSM	ce ; ESC × 1 ESC [> 1 h ; Enable 25th line ; ESC × 2 ESC [> 2 h ; No keyboard click ; ESC × 3 ESC [> 3 h ; Enter hold screen mode
30 31 32 33 0911 34 0912 35 36 0914 37 0915 38 39 0917 40 0918 41 42 0918 44	01 0B0B 02 0B2D 03 0B3C 04 0B33	SMST	EQU DB DW DB DW DB DW DB DW	SC L > Pn h sequen \$ 1 E25L 2 NKC 3 EHSM 4 SBC	ce ; ESC × 1 ESC [> 1 h ; Enable 25th line ; ESC × 2 ESC [> 2 h ; No keyboard click ; ESC × 3 ESC [> 3 h ; Enter hold screen mode ; ESC × 4 ESC [> 4 h ; Set "block" cursor
30 31 32 33 0911 34 0912 35 36 0914 37 0915 38 0917 40 0918 41 42 091A 43 091B 44 45 091D	01 0B0B 02 0B2D 03 0B3C 04 0B33	SMST	EQU DB DW DB DW DB DW DB DW	SC L > Pn h sequen \$ 1 E25L 2 NKC 3 EHSM 4 SBC	<pre> ; ESC × 1 ESC [> 1 h ; Enable 25th line ; ESC × 2 ESC [> 2 h ; No keyboard click ; ESC × 3 ESC [> 3 h ; Enter hold screen mode ; ESC × 4 ESC [> 4 h ; Set "block" cursor ; ESC × 5 ESC [> 5 h </pre>
30 31 32 33 0911 34 0912 35 36 0914 37 0915 38 39 0917 40 0918 41 42 0918 44	01 0B0B 02 0B2D 03 0B3C 04 0B33 05 055	smst	EQU DB DW DB DW DB DW	\$ 1 E25L 2 NKC 3 EHSM 4 SBC 5 DC	ce ; ESC × 1 ESC [> 1 h ; Enable 25th line ; ESC × 2 ESC [> 2 h ; No keyboard click ; ESC × 3 ESC [> 3 h ; Enter hold screen mode ; ESC × 4 ESC [> 4 h ; Set "block" cursor
30 31 32 33 0911 34 0912 35 36 0914 37 0915 38 39 0917 40 0918 41 42 0918 43 0918 44 45 0918 46 091E 47	01 0B0B 02 0B2D 03 0B3C 04 0B33 05 055	SMST	EQU DB DW DB DW DB DW DB DW DB DW	\$ 1 E25L 2 NKC 3 EHSM 4 SBC 5 DC	FSC × 1 ESC [> 1 h Finable 25th line FSC × 2 ESC [> 2 h No keyboard click FSC × 3 ESC [> 3 h Finter hold screen mode FSC × 4 ESC [> 4 h Set "block" cursor FSC × 5 ESC [> 5 h Disable cursor
30 31 32 33 0911 34 0912 35 36 0914 37 0915 38 39 0917 40 0918 41 42 0918 43 0918 44 45 0910 46 091E 47 48 0920	01 0B0B 02 0B2D 03 0B3C 04 0B33 05 0B5F	SMST	EQU DB DW DB DW DB DW DB DW DB DW DB DW	SC L > Pn h sequen \$ 1 E25L 2 NKC 3 EHSM 4 SBC 5	<pre> ; ESC × 1 ESC [> 1 h ; Enable 25th line ; ESC × 2 ESC [> 2 h ; No keyboard click ; ESC × 3 ESC [> 3 h ; Enter hold screen mode ; ESC × 4 ESC [> 4 h ; Set "block" cursor ; ESC × 5 ESC [> 5 h ; Disable cursor ; ESC × 6 ESC [> 6 h </pre>
30 31 32 33 0911 34 0912 35 36 0914 37 0915 38 39 0917 40 0918 41 42 0918 43 0918 44 45 0918 46 091E 47	01 0B0B 02 0B2D 03 0B3C 04 0B33 05 055	SMST	EQU DB DW DB DW DB DW DB DW DB DW	\$ 1 E25L 2 NKC 3 EHSM 4 SBC 5 DC	FSC × 1 ESC [> 1 h Finable 25th line FSC × 2 ESC [> 2 h No keyboard click FSC × 3 ESC [> 3 h Finter hold screen mode FSC × 4 ESC [> 4 h Set "block" cursor FSC × 5 ESC [> 5 h Disable cursor
30 31 32 33 0911 34 0912 35 36 0914 37 0915 38 39 0917 40 0918 41 42 0918 43 0918 44 45 0918 46 0916 47 0916 48 0920 49 0921 50 51 0923	01 0B0B 02 0B2D 03 0B3C 04 0B33 05 0B5F	SMST	EQU DB DW DB DW DB DW DB DW DB DW DB DW	\$	SESC × 1 ESC [> 1 h SENT ENT ENT ENT ENT ENT ENT ENT ENT ENT
30 31 32 33 0911 34 0912 35 36 0914 37 0915 38 39 0917 40 0918 41 42 0918 43 0918 44 45 0918 46 0916 47 0916 48 0920 49 0921 50 51 0923	01 0B0B 02 0B2D 03 0B3C 04 0B33 05 05 0B5F	SMST	EQU DB DW	\$ 1.	SESC × 1 ESC [> 1 h SENT ESC × 2 ESC [> 2 h SESC × 2 ESC [> 2 h SESC × 3 ESC [> 3 h SESC × 3 ESC [> 3 h SESC × 4 ESC [> 4 h SESC × 4 ESC [> 4 h SESC × 5 ESC [> 5 h Disable cursor SESC × 6 ESC [> 6 h SESC × 6 ESC [> 6 h SESC × 7 ESC [> 7 h
30 31 32 33 0911 34 0912 35 36 0914 37 0915 38 39 0917 40 0918 41 42 091A 43 091B 44 45 091D 46 091E 47 48 0920 49 0921 50 51 0923 52 0924 53	01 0B0B 02 0B2D 03 0B3C 04 0B33 05 0B5F 06 0B48 07 0B4E	SMST	EQU DB DW DB DW DB DW DB DW DB DW DB DW	\$	SESC × 1 ESC [> 1 h SENT ENT ENT ENT ENT ENT ENT ENT ENT ENT
30 31 32 33 0911 34 0912 35 36 0914 37 0915 38 39 0917 40 0918 41 42 091A 43 091B 44 45 091D 46 091E 47 48 0920 49 0921 50 51 0923 52 0924 53	01 0B0B 02 0B2D 03 0B3C 04 0B33 05 0B5F 06 0B48 07 0B4E	SMST	EQU DB DW	SC L > Pn h sequen \$ 1 E25L 2 NKC 3 EHSM 4 SBC 5 DC 6 EKSM 7 EKAM	; ESC × 1 ESC [> 1 h ; Enable 25th line ; ESC × 2 ESC [> 2 h ; No keyboard click ; ESC × 3 ESC [> 3 h ; Enter hold screen mode ; ESC × 4 ESC [> 4 h ; Set "block" cursor ; ESC × 5 ESC [> 5 h ; Disable cursor ; ESC × 6 ESC [> 6 h ; Enter keypad shifted mode ; ESC × 7 ESC [> 7 h ; Enter keypad alternate mode
30 31 32 33 0911 34 0912 35 36 0914 37 0915 38 39 0917 40 0918 41 42 091A 43 091B 44 45 091B 47 091B 48 0920 49 0921 50 51 0923 52 0924 53 54 0926	01 0B0B 02 0B2D 03 0B3C 04 0B33 05 0B5F 06 0B48 07 0B4E	SMST	EQU DB DW	\$	FSC × 1 ESC [> 1 h FEND × 2 ESC [> 2 h No keyboard click FESC × 3 ESC [> 3 h Enter hold screen mode FESC × 4 ESC [> 4 h Set "block" cursor ESC × 5 ESC [> 5 h Disable cursor FESC × 6 ESC [> 6 h Enter keypad shifted mode FESC × 7 ESC [> 7 h Enter keypad alternate mode
30 31 32 33 0911 34 0912 35 36 0914 37 0915 38 39 0917 40 0918 41 42 091A 43 091B 44 45 091D 46 091E 47 48 0920 49 0921 50 51 0923 52 0924 53	01 0B0B 02 0B2D 03 0B3C 04 0B33 05 0B5F 06 0B48 07 0B4E	SMST	EQU DB DW	SC L > Pn h sequen \$ 1 E25L 2 NKC 3 EHSM 4 SBC 5 DC 6 EKSM 7 EKAM	; ESC × 1 ESC [> 1 h ; Enable 25th line ; ESC × 2 ESC [> 2 h ; No keyboard click ; ESC × 3 ESC [> 3 h ; Enter hold screen mode ; ESC × 4 ESC [> 4 h ; Set "block" cursor ; ESC × 5 ESC [> 5 h ; Disable cursor ; ESC × 6 ESC [> 6 h ; Enter keypad shifted mode ; ESC × 7 ESC [> 7 h ; Enter keypad alternate mode

** Z-29 COMPU	JTER TERMINAL FIRE	ASSEMBLER (V1)-218 7 MWARE VERSION 1.03 **	¥*			
1 092A	OB62		DW	EACR	; Enable auto CR on LF	
2 3	0009	SMSTL	EQU	(\$-SMST)/3	; Table length	
• • • • • • • • • • • • • • • • • • • •						
				•••••		

)

)

)

)

)

)

Z29R0M	; MICROBI 29.COMPU	ENCH 805	51 CROSS MINAL.FI	ASSEMBLE	R (V1)-218 2 RSION 1,03 **	4-FEB-83 15 *	:30:06 PAGE 80		••••
••••••	1 2	• • • • • • • • • • • • • • • • • • • •				RM\$	reset mode sequence		
	2 3 4 5				7 	*RMS* i sequeni	resets the mode specifi ce.	ed.by.the.last_character_in_the	
······································	6 7				.	ENTRY	none		
10	9 					EXIT	none		
1	1					USES	all		
	3 4 092C	11	2A	••••••	RMS:	ACALL	SETDISP	; Set dispatch and return	
	5 092E	11	2F	• • • • • • • • • • • • • • • • • • • •		ACALL	SETNORM	; Restore dispatch address	
18	7 0930 3 0931	94	31			CLR SÜBB	C A, #′1′	; Subtract offset	
20	9.0933 0.0935	40 	OE			JC CJNE	RMSRET A, #('9'-'0'), RMS1		
23	1 0938 2 093A 3 093B	50 	09 F0	09	RMS1:	JNC INC MOV	RMSRET A B, #RMSTL	; Check for upper bounds ; Bump up into range	
24	1 093E	75 90 01	0956 36	· · · · · · · · · · · · · · · · · · ·		MÖV AJMP	DPTR, #RMST STJMP	; (B) = table length ; (DPTR) = ESC y Pn table ; Jump to any routine	
28 27	5 0941 5 7 0943	22						, odine to any roditive	• • • • • • • • • • • • • • • • • • • •
		*****			RMSRET:	RET			
					RMSRET:	RET	•••••••••••••••••••••••••••••••••••••••		
					RMSRET:	RET			
					RMSRET:	RET			
					RMSRET:	RET			
					RMSRET:	RET			
					RMSRET:	RET			
					RMSRET:	RET			
					RMSRET:	RET			
					RMSRET:	RET			
					RMSRET:	RET			
			••••••						
			••••••						

	; ;	A2RM -	ANSI mode #2 reset	mode sequence
	,		reset the specified	
			reset the sreetlitet	T. MYTT T. MT. T
		ENTRY	none	

	; ;	EXIT	none	
	*	USES	A, B, DPTR	
			CNE	. C. L
0944 31 D1 0946 40 FB	A2RM:	ACALL.	GNP RMSRET	; Get next (or first) parameter ; IF parameter THEN
0948 75 F0	09	MOV	B, #RMSTL	; (B) = table length
0948 90 0956	¥7	MOV	DPTR, #RMST	; (DPTR) = ESC [> Pn table
094E 31 F7		ACALL	XSTAB	; Search table
0950 40 F1		Jč	RMSRET	; IF found THEN
0952 11 3A		AÇALL.	STCALL	; jump to routine
0952 11 3A 0954 80 EE		SJMP	A2RM	; and keep soins
		PMST -	- ANSI reset mode se	mence table
	3 3			
	*			
	,			meters for ESC [> Pn] sequence.
		RMST	contains valid mare	
0956	RMST	*RMST* EQU		
0956 01	; ; RMST	*RMST* EQU DB	contains valid para \$	meters for ESC [> Pn] sequence. ; ESC Y 1 ESC [> 1]
	; ; RMST	*RMST* EQU	contains valid para	meters for ESC [> Pn] sequence.
0956 01 0957 0B15	; ; RMST	*RMST* EQU DB DW	contains valid para \$ 1 D25L	meters for ESC [> Pn] sequence. ; ESC Y 1 ESC [> 1] ; Disable 25th line
0956 01 0957 0815 0959 02 095A 0826	; ; RMST	*RMST* EQU DB	contains valid para \$	meters for ESC [> Pn] sequence. ; ESC Y 1 ESC [> 1]
0956 01 0957 0815 0959 02 095A 0826	; ; RMST	*RMST* EQU DB DW DB	\$ 1 D25L EKC	meters for ESC [> Pn] sequence. ; ESC y 1
0956 01 0957 0815 0959 02 095A 0826	; ; RMST	*RMST* EQU DB DW DB	contains valid para \$ 1 D25L	meters for ESC [> Po] sequence. ; ESC y 1 ESC [> 1] ; Disable 25th line. ; ESC y 2 ESC [> 2]
0956 01 0957 0B15 0959 02 095A 0B26 095C 03 095D 0B43	RMST	*RMST* EQU DB DW DB DW DB	\$ 1 D25L 2 EKC 3 XHSM	meters for ESC [> Po] sequence. ; ESC Y 1
0956 01 0957 0815 0959 02 095A 0826 095C 03 095D 0843	RMST	*RMST* EQU DB DW DB DW DB	s 1 D25L 2 EKC 3 XHSM	meters for ESC [> Po] sequence. ; ESC y 1
0956 01 0957 0815 0959 02 095A 0826 095C 03 095D 0843 095F 04 0960 0837	RMST	*RMST* EQU DB DW DB DW	\$ 1 D25L 2 EKC 3 XHSM	meters for ESC [> Pn] sequence. ; ESC Y 1
0956 01 0957 0B15 0959 02 095A 0B26 095C 03 095D 0B43 095F 04 0960 0B37	RMST	#RMST* EQU DB DW DB DW DB DW DB DW DB DW	\$ 1 D25L 2 EKC 3 XHSM 4 SUC	meters for ESC [> Po] sequence. ; ESC Y 1
0956 01 0957 0815 0959 02 095A 0826 095C 03 095D 0843 095F 04 0960 0837	RMST	#RMST* EQU DB DW DB DW DB DW	s contains valid para \$ 1	meters for ESC [> Pn] sequence. ; ESC Y 1
0956 01 0957 0815 0959 02 095A 0826 095C 03 095D 0843 095F 04 0960 0837	RMST	*RMST* EQU DB DW DB DW DB DW DB DW DB DW DB DW	s 1 D25L 2 EKC 3 XHSM 4 SUC 5 EC	######################################
0956 01 0957 0815 0959 02 095A 0826 095C 03 095D 0843 095F 04 0960 0837 0962 05 0963 085C 0965 06	RMST	#RMST* EQU DB DW DB DW DB DW DB DW DB DW	\$ 1 D25L 2 EKC 3 XHSM 4 SUC	meters for ESC [> Po] sequence. ; ESC Y 1
0956 01 0957 0B15 0959 02 095A 0B26 095C 03 095D 0B43 095F 04 0960 0B37 0962 05 0963 0B5C 0965 06	RMST	*RMST* EQU DB DW DB DW DB DW DB DW DB DW DB DW	s 1 D25L 2 EKC 3 XHSM 4 SUC 5 EC	######################################
0956 01 0957 0815 0959 02 095A 0826 095C 03 095D 0843 095F 04 0960 0837 0962 05 0963 085C 0965 06 0966 0848	RMST	*RMST* EQU DB DW DB DW	s 1 D25L 2 EKC 3 XHSM 4 SUC 5 EC	meters for ESC [> Po] sequence. ; ESC y 1
0956 01 0957 0815 0959 02 0958 0826 0950 03 0950 0843 095F 04 0960 0837 0962 05 0963 085C 0965 06 0966 0848	RMST	#RMST* EQU DB DW DB DW	s 1 D25L 2 EKC 3 XHSM 4 SUC 5 EC 4 XKSM	######################################
0956 01 0957 0815 0959 02 095A 0826 095C 03 095D 0843 095F 04 0960 0837 0962 05 0963 085C 0965 06 0966 0848	RMST	*RMST* EQU DB DW DB DW	s 1 D25L 2 EKC 3 XHSM 4 SUC 5 EC	######################################

Z29ROM; MICROBENCH 8051 CROSS *** Z-29 COMPUTER TERMINAL FIR	ASSEMBLER (V1)-218 24 MWARE YERSION 1,03 ***	-FEB-83 15:30:06 PAGE 82		
2 0009		•		
2 0009	RMSTL	EQU (\$-RMST)/3	; Table length	
		••••••	••••••	••••••
••••••			•••••	
		•••••		
	•••••		• • • • • • • • • • • • • • • • • • • •	
·····				
		• • • • • • • • • • • • • • • • • • • •		
•••••	•••••			
		•••••••••••••••••••••••••••••••••••••••		••••••
	•••••••••••••••••••••••••••••••••••••••	••••••	•••••	
	••••••	• • • • • • • • • • • • • • • • • • • •		
		•••••	•••••	
•••••••••••••••••••••••••••••••••••••••	•••••			••••••
•				
	•••••			
		·		•••••
•••••		•••••		
***************************************	•••••		••••••	
•••••	•••••		***************************************	
•••••	• • • • • • • • • • • • • • • • • • • •			
***************************************	••••			
			••••••	
	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •

C

1						PSD -	parameter string decoder	
2	•••••				9			of decimal numbers separated
		• • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		by a s	emicolon (to a maximum of	8) until the final character
5								imal and not a semicolon) is
6					3	inpute	d.	
······································								······································
9 10						ENTRY	(A) = next parameter ch	
11						EXIT	<pre>(A) = last character in 'C' = set if done decod</pre>	Puted (Ps usually)
12 13					; ;		'C' = set 1+ done decod	ing, clear etherwise
14 15				•••••	,	ÚSÉS	A, B, DPTR, TEMP	
16			4.0	oF	PSD:	JB	PSDF, PSDB	; IF first time through THEN
17.09 18.09		20 D2	<u>18</u>	OE		SETB	PSDF	; tossle flas
19 09		. 75 	1A 19	F8		MOV	PMADRL, #LOW (PMBUF)	; set parameter pointer
20 09	79		19	57		MOV	PMADRH, #HIGH (PMBUF)	
21.09 22	?7C	75	1B	FF		MOV	PMNUM, #-1	; init number of parameters
	77F	.75	1C	00		MOV	PMVALUE, #0	; init parameter value
23 09 24					5055	C INC	A WZOZII DODDI	
25 09 26 09	782 	B4 50	3A 1B	00	PSDB: PSDB1:	CJNE JNC	A, #′9′+1, PSDB1	; IF below upper range AND
27 09		.B4	30	00	1 3001.	CUNE	A, #101, PSDB2	,
28 09	8A	40	16		PSDB2:	JC	PSDE	; IF above lower ranse THEN
27		94	30			SUBB	Α, #′0′	; remove ASCII offset
30 09 31 09		24 C5	1C			XCH	A, PMVALUE	; set previous value
32 09			F0	ÖÄ		MOV	B, #10	
33 09	993	A4				MUL	AB	; value := value * 10 ; value := value + newparm
34 09		25	1C			ADD	A, PMVALUE	; value := value + newparm
35.09		. F5 E5	1C			MOV MOV	PMVALUE, A A, PMNUM	; save new parameter value
36 09 37 09		E5 B4	IB EE	0.3		CUNE	A, #-1, PSD1	; IF parmnum = -1 THEN
379? 38 09		 75	FF	<u>03</u>		MOV	PMNUM, #0	; parmnum := O
39.09	PAQ				PSD1:	.CLR	C	; flas = still parsins strins
40 09	7A1	22				RET		
41 42 09	 A2	 85	1A	82	PSDE:	MOV	DPL, PMADRL	; ELSE get parameter address
43.09			19	83		MOV	DPH, PMADRH	
44 09		FD				MOV MOV	TEMP, A A, PMVALUE	; save character temporary ; get current parameter value
45.09 46.09		<u>E</u> 5	1Ç			MOVX	@DPTR, A	; save in parameter buffer
47 09		. A3				INC	DPTR	; bump parameter address
48 09		E5	82			MOV	A, DPL	; set low byte address to
49.09		B4	00	05 08		CUNE	A, #LOW(PMBUF+8), PSD2	; IF overflow THEN ; give max number of parameters
50 09		75	1B	08		MOV SJMP	PMNUM, #8 PSD3	; give max number of Parameters ; and skip
51 09 52 09		80 85	08 82	1A	PSD2:	MOV	PMADRL, DPL	; ELSE replace
52 09 53 09		. 85	83	19	r waran *	MOV	PMADRH, DPH	; parameter address
54 09	PBD	05	18			INC	PMNUM	; bump number of parameters
5509	9.BF	ED			PSD3:	MQY	A, TEMP	; restore original character.
	2C0	B4	3B	oc		CJNE	A, #';', PSDRET	; IF ';' THEN

.....

	1 0906	ΑΠ	1 B		MOV	TEMP	PMNIIM		
• • • • • • • • • • • • • • • • • • • •	2 09C8	AD	1 <u>B</u>	02	CUNE	!5:05.? TEMP,	PMNUM #0, PSD4		check for ';' as first char
	3.09CB. 4.09CD	05 	1B		INC CLR	PMNUM C			IF so THEN adjust number flas = still parsins strins
	4 09CD 5 09CE	27		PSD4:	CLR	С			
	6 09CF	D3		PSDRET:	RET	c	•••••	· · · · · · · · · · · · · · · · · · ·	Flas = finished parsing
	7.09DO	22			RET				
					• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
•••••			• • • • • • • • • • • • • • • • • • • •	•••••				• • • • • • • • • • • • • • • • • • • •	•••••••••••••••••••••••••••••••••••••••
			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •				
						• • • • • • • • • • • • • • • • • • • •			
• • • • • • • • • •									
			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •				••••••
									••••••
		• • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •				••••••
				• • • • • • • • • • • • • • • • • • • •					·····
			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			••••••	• • • • • • • • • • • • • • • • • • • •	•••••••••••••••••••••••••••••••••••••••
• • • • • • • • • • • •		• • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	
		• • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
• • • • • • • • • •	• • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •					• • • • • • • • • • • • • • • • • • • •	
			• • • • • • • • • • • • • • • • • • • •						
• • • • • • • • • •	••••••		• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •				
		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •					
					• • • • • • • • • • • • • • • • • • • •				

C

C C

<i></i> 		,				GNP -	set next parameter	
2 3					;	*GNP* :	sets the next (or firs	t) parameter from the parameter
4 5					,	buffer	•	
 6 7				• • • • • • • • • • • • • • • • • • • •		ENTRY	none	
						EXIT		b) parameter. O if no more
<u>9</u>							C' = set if no Pare	t) parameter, O if no more ameters left, clear otherwise
11						ÜSES	A, DPTR	
13 14 15 09	To 4		10	08	GNP:	JB	GNPF, GNP1	; IF first time since ESC THEN
16 09	L!! D4	.20 D2	19			SETB	GNPF	; IF first time since ESC THEN ; toggle flag
17 09	D6	75 75	1A	F8 57		MOV	PMADRL, #LOW (PMBUF)	
18 09	D9	75	19			MOV	PMADRH, #HIGH (PMBUF	-)
19 20 09:	nc:	 85	1A	 82	GNP1:	MOV	DPL, PMADRL	; Get pointer to next parameter
2102	DF	.85	1.9	83		MOV	DPH: PMADRH	; Get number of parameters 07
22 09	E2	E5	1B			MOV	A, PMNUM	; Get number of Parameters U/ ; IF no more or none at all
2399. 24. 09	年4	.79 D3	92			JŅZ SETB	GNP2	; 'C' = no more left
24 09 25 09	F7	.22				RET		
26 09	E8	E0			GNP2:	MOVX	A, @DPTR	; ELSE set parameter
						INC	DPTR	; bump pointer
2799	'E?	.A3	<u></u>					
28 09	EA	85	82	1A		MOV	PMADRL, DPL	; and save
28 09 29 09	EA ED	. 85 . 85	83	1A 19		MOV MOV DEC		; and save ; dec number of parameters
28 09 29 09 30 09	EA ED FO	85 .85 15		1A 19		MOV MOV DEC CLR	PMADRL, DPL PMADRH, DPH	; and save
28 09 29 09	EA ED FO F2	. 85 . 85	83	1A 19		MOV MOV DEC	PMADRL, DPL PMADRH, DPH PMNUM	; and save ; dec number of parameters
28 09 29 09 30 09 31 09	EA ED FO F2	85 .85 .15 .C3	83	1A 19		MOV MOV DEC CLR	PMADRL, DPL PMADRH, DPH PMNUM	; and save ; dec number of parameters
28 09 29 09 30 09 31 09	EA ED FO F2	85 .85 .15 .C3	83	1A 19		MOV MOV DEC CLR	PMADRL, DPL PMADRH, DPH PMNUM	; and save ; dec number of parameters
28 09 29 09 30 09 31 09	EA ED FO F2	85 .85 .15 .C3	83	1A 19		MOV MOV DEC CLR	PMADRL, DPL PMADRH, DPH PMNUM	; and save ; dec number of parameters
28 09 29 09 30 09 31 09	EA ED FO F2	85 .85 .15 .C3	83	1A 19		MOV MOV DEC CLR	PMADRL, DPL PMADRH, DPH PMNUM	; and save ; dec number of parameters
28 09 29 09 30 09 31 09	EA ED FO F2	85 .85 .15 .C3	83	1A 19		MOV MOV DEC CLR	PMADRL, DPL PMADRH, DPH PMNUM	; and save ; dec number of parameters
28 09 29 09 30 09 31 09	EA ED FO F2	85 .85 .15 .C3	83	1A 19		MOV MOV DEC CLR	PMADRL, DPL PMADRH, DPH PMNUM	; and save ; dec number of parameters
28 09 29 09 30 09 31 09	EA ED FO F2	85 .85 .15 .C3	83	1A 19		MOV MOV DEC CLR	PMADRL, DPL PMADRH, DPH PMNUM	; and save ; dec number of parameters
28 09 29 09 30 09 31 09	EA ED FO F2	85 .85 .15 .C3	83	1A 19		MOV MOV DEC CLR	PMADRL, DPL PMADRH, DPH PMNUM	; and save ; dec number of parameters
28 09 29 09 30 09 31 09	EA ED FO F2	85 .85 .15 .C3	83	1A 19		MOV MOV DEC CLR	PMADRL, DPL PMADRH, DPH PMNUM	; and save ; dec number of parameters
28 09 29.09 30 09 31.09	EA ED FO F2	85 .85 .15 .C3	83	1A 19		MOV MOV DEC CLR	PMADRL, DPL PMADRH, DPH PMNUM	; and save ; dec number of parameters
28 09 29 09 30 09 31 09	EA ED FO F2	85 .85 .15 .C3	83	1A 19		MOV MOV DEC CLR	PMADRL, DPL PMADRH, DPH PMNUM	; and save ; dec number of parameters
28 09 29 09 30 09 31 09	EA ED FO F2	85 .85 .15 .C3	83	1A 19		MOV MOV DEC CLR	PMADRL, DPL PMADRH, DPH PMNUM	; and save ; dec number of parameters
28 09 29 09 30 09 31 09	EA ED FO F2	85 .85 .15 .C3	83	1A 19		MOV MOV DEC CLR	PMADRL, DPL PMADRH, DPH PMNUM	; and save ; dec number of parameters
28 09 29 09 30 09 31 09	EA ED FO F2	85 .85 .15 .C3	83	1A 19		MOV MOV DEC CLR	PMADRL, DPL PMADRH, DPH PMNUM	; and save ; dec number of parameters
28 09 29 09 30 09 31 09	EA ED FO F2	85 .85 .15 .C3	83	1A 19		MOV MOV DEC CLR	PMADRL, DPL PMADRH, DPH PMNUM	; and save ; dec number of parameters
28 09 29 09 30 09 31 09	EA ED FO F2	85 .85 .15 .C3	83	1A 19		MOV MOV DEC CLR	PMADRL, DPL PMADRH, DPH PMNUM	; and save ; dec number of parameters
28 09 29 09 30 09 31 09	EA ED FO F2	85 .85 .15 .C3	83	1A 19		MOV MOV DEC CLR	PMADRL, DPL PMADRH, DPH PMNUM	; and save ; dec number of parameters
28 09 29 09 30 09 31 09	EA ED FO F2	85 .85 .15 .C3	83	1A 19		MOV MOV DEC CLR	PMADRL, DPL PMADRH, DPH PMNUM	; and save ; dec number of parameters
28 09 29 09 30 09 31 09	EA ED FO F2	85 .85 .15 .C3	83	1A 19		MOV MOV DEC CLR	PMADRL, DPL PMADRH, DPH PMNUM	; and save ; dec number of parameters

)

•

)

3

)

•

Z29ROM; MICROBENCH 8051 CROSS *** Z-22.COMPUTER TERMINAL FIR	ASSEMBLER (V1)-218 24-	-FEB-83 15:30:06 PAGE 86
1		XBLINAD - build line address
3 4 5 6	; ; ;	*XBLINAD* is a jump to the routine *BLINAD*. It is used to save bytes so that routines may use a short jump to here instead of a long jump to the other routine.
	XBL I NAD:	LJMP BLINAD
		XSTAB - search table
15 16 17 18	,	*XSTAB* is a jump to the routine *STAB*. It is used to save bytes so that routines may use a short jump to here instead of a long jump to the other routine.
19 20 09F7 02 013A	XSTAB:	LUMP STAB
	•••••	
	•••••	······································
	•••••	
		······

Z2'	9ROM; MICROI * Z-29 COMPI	UTER TERM	MINAL FIF	RMWARE VER	SION 1.03 ***			
	1				* *	APCA -	ANSI perform cursor addr	essina
	2 3				;	*APCA*	checks to see that the 1	ine and column values are legal
	4				*	and set	s the cursor to the requ	ested address. If the values number remains and/or the cursor
	5					goes to	the last column on the	line.
	7							
	9				, ;	ENTRY	none	
	11					EXIT	none	
	11				🛊			
	1314					USES	all	
	15							
	16 09FA	31 60	DI		APCA:	JZ	GNP ACA2	; Get next parameter ; IF non-zero THEN
	17 09FC 18 09FE	14	OA			DEC	Ä	; bring into range 024
	19 09FF 20 0A02	B4 	18 15	<u>04</u>		CJNE JNB	A, #MAXLINE, ACA1 L25EN, ACA3	; check range ; IF line 25 disbld THEN skip
	20 0A02 21 0A05	D3		03		SETB	С	; trick ckeck below
	22 0A06	5ö F5	62		ACA1:	JNC	ACA3 YPOS, A	; IF in range THEN ; set Y-position
	23 0A08 24		12		ACA2:	MOV		
	25 0A0A	31	D1		ACA3:	ACALL	GNP ACA5	; Get next parameter ; IF non-zero THEN
	26 0A0C 27 0A0E	60 14				DEC	A	; bring into range 079
	28 0A0F	B4	50	00		CUNE JC	A, #MAXCHAR+1, ACA3A ACA5	; check range ; IF not in range THEN
	29 0A12	40	02		ACA3A:	CHC.	HUHU	
		74	4F			MOV	A, #MAXCHAR	; go to end of line
	30 0A14 31 0A16	74 F5	4F 11		ACA5:	MOV	XPOS, A	; go to end of line ; Set X-position
	30 0A14	74						; go to end of line
	30 0A14 31 0A16	74 F5	11			MOV	XPOS, A	; go to end of line ; Set X-position
	30 0A14 31 0A16	74 F5	11			MOV	XPOS, A	; go to end of line ; Set X-position
	30 0A14 31 0A16	74 F5	11			MOV	XPOS, A	; go to end of line ; Set X-position
	30 0A14 31 0A16	74 F5	11			MOV	XPOS, A	; go to end of line ; Set X-position
	30 0A14 31 0A16	74 F5	11			MOV	XPOS, A	; go to end of line ; Set X-position
	30 0A14 31 0A16	74 F5	11			MOV	XPOS, A	; go to end of line ; Set X-position
	30 0A14 31 0A16	74 F5	11			MOV	XPOS, A	; go to end of line ; Set X-position
	30 0A14 31 0A16	74 F5	11			MOV	XPOS, A	; go to end of line ; Set X-position
	30 0A14 31 0A16	74 F5	11			MOV	XPOS, A	; go to end of line ; Set X-position
	30 0A14 31 0A16	74 F5	11			MOV	XPOS, A	; go to end of line ; Set X-position
	30 0A14 31 0A16	74 F5	11			MOV	XPOS, A	; go to end of line ; Set X-position
	30 0A14 31 0A16	74 F5	11			MOV	XPOS, A	; go to end of line ; Set X-position
	30 0A14 31 0A16	74 F5	11			MOV	XPOS, A	; go to end of line ; Set X-position
	30 0A14 31 0A16	74 F5	11			MOV	XPOS, A	; go to end of line ; Set X-position
	30 0A14 31 0A16	74 F5	11			MOV	XPOS, A	; go to end of line ; Set X-position
	30 0A14 31 0A16	74 F5	11			MOV	XPOS, A	; go to end of line ; Set X-position
	30 0A14 31 0A16	74 F5	11			MOV	XPOS, A	; go to end of line ; Set X-position

Z29ROM; MICROBEN	CH 8051 CROSS ASSEMBLER	(V1)-218 24-FEB-83 15	5:30:06 PAGE 88	
	R.TERMINAL.F.IRMWARE.VER		erase in display	
3 3		7	erases the amount of the	screen specified by Pn.
5		; .; ; ENTRY		
		; ENIRY	none	
	••••••	; USES	all	
11				
13.0A1A 14.0A1C 15.0A1E	70 02	FID: AÇALL JNZ AJMP	EID1	; Get parameter ; IF zero THEN
16			A	; erase to eos and return
19. 0A23	70 02 61F1	JNZ	EID2	; IF one THEN ; erase from beginning & ret
	1 4 70 02		A EIR	; IF two THEN
23.0A28 24	61AC	AJMP	CLRS	.ierase.screen and return
25 . 9.628	22	EIR:RET		
	••••••			
	•			
		•••••		
		• • • • • • • • • • • • • • • • • • • •		
	•••••			
•••••		• • • • • • • • • • • • • • • • • • • •	•••••	

				EIL -	erase in line	
			; ;	*EIL*	erases the portio	n of the current line as specified
			# #	by Pn.		
				ENTRY	none	
· · · · · · · · · · · · · · · · · · · ·				EXIT	none	
) 			, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	USES	all	
3						
OA2B	3i	<u>Dī</u>	EIL:	ACALL	GNP	; Get parameter : IF zero THEN
0A2D 0A2F	<u>70</u>	<u>02</u>		JNZ AJMP	EIL1 EOL	; IF zero THEN ; erase to eol and return
•	14		EIL1:	DEC	A	
0A31 0A32	14 70	02 CB	EILI.	JNZ AJMP	EIL2 EBL	; IF one THEN ; erase from beginning & ret
0A34	70 61	СВ				; erase from beginning & ret
2 0A36	14		EIL2:	DEC	A	. IC A THEN
0A37	<u>70</u>	F1		JNZ AJMP	EIR EEL	; IF two THEN ; erase entire line and ret

	1		. 		; ;	ASGM -	ANSI set eraphics m	ode
	2				;			A84
*	্ব					*ASGM*	.sets.on.nesets.any	or.all.of.the.attributes.and.
	5				7	araphi	cs modes.	
	6		• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
	.7	· • • • • • • • • • • • • • • • • • • •			5	ENTRY	none	
	~				7			
	. Y	• • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		EXI.T	n.o.n.e	••••••
					,			
				• • • • • • • • • • • • • • • • • • • •			¢.()	••••••
		<u>.</u>						
	14 OA3B	31	D1		ASGM:	ACALL		; Get first parameter
	16 OA3D	75	F0	07	ASG1:			
		9 0	9A50		H201•	MQV MQY	B, #ASGMTL DPTR, #ASGMT	<pre>; (B) = table lensthi.(DPTR) = set mode table</pre>
	18 0A43	31	F7			ACALL	XSTAB	: Search table
	L90A45	<u>4</u> Ω	92			JQ	ASG0	
	20 0A47	51	4E			ACALL	ASGO	; call the couting
	449 04 7 22 0A4B	31 50	FO	• • • • • • • • • • • • • • • • • • • •	ASGQ:		GNP	j.Get_next_parameter
						OING	HOUI	i It done IMEN return
2	24			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	i ELSE keep looping
		E4			A\$GQ:	CLR	A	
	26 OA4F	73						
					•••••	•••••	@A+DPTR	; Jump to routine
			• • • • • • • • • • • • • • • • • • • •				@A+DPTR	; Jump to routine
							@A+DPTR	; Jump to routine
							@A+DPTR	; Jump to routine
							@A+DPTR	; Jump to routine
							@A+DPTR	; Jump to routine
							@A+DPTR	; Jump to routine
							@A+DPTR	; Jump to routine
							@A+DPTR	; Jump to routine
							@A+DPTR	; Jump to routine
							@A+DPTR	; Jump to routine
							@A+DPTR	; Jump to routine
							@A+DPTR	; Jump to routine
							@A+DPTR	; Jump to routine
							@A+DPTR	; Jump to routine
							@A+DPTR	; Jump to routine
							@A+DPTR	; Jump to routine
							@A+DPTR	; Jump to routine
							ea+DPTR	; Jump to routine
							ea+DPTR	; Jump to routine

(((

CC

1			ASGMT	- ANSI set graphics	mode table
3 4			*ASGM	T* contains the addre	ess of the routines which set the aphic mode.
					hird character of the ANSi escape
6 8		<u>.</u>	seque	entries are: The tr nce followed by the a equested function.	address of the routine which performs
9 10		,	the r		
11	0A50	ASGMT	EQU	\$	
12 13 0A50	00		DB DW	O XATR	; ESC [O m
14 0A5:	0F26			XATR	; Exit all attributes
15 16 0A5	3 02		DB	2	; ESC [2 m
17 0A54 18			DW	EHALF	; Enter half intensity mode
19 0A5	5 04		DB DW	4 EUNDL	; ESC [4 m
20 0A5	7 OF2D		DW	EUNDL	; Enter underline mode
21 22 0A51	 9 05		DB	5	; ESC [5 m
23.0A56	90F30		Þ₩	EBLNK	; Enter blink mode
_24 _25_9 <u>45</u> !	07		DB.	7 ERVM	: ESC [7 m : Enter reverse video mode
26 0A5	0 OF33		DW	ERVM	
27 28 0A5I			DB	10	; ESC [10 m
29. 0A6	O OF39		DW	EGM	; Enter graphics mode
30 31 086	2QB		DB	11	; ESC [11 m ; Exit graphics mode
32 OA6			DW	XGM	; Exit graphics mode
33 34	0007	ASGMTL	EQU	(\$-ASGMT)/3	; Table length
	····				

Z29R	OM; MICRO	BENCH 80	51 CROSS	ASSEMBL	ER (V1)-218 2	4-FEB-83 15	5:30:06 PAGE 92		
•••••	1	······	·······················		ERSION 1.03 **		- ANSI set protected	fields	
	2				;				•••••
• • • • • • • • • • • • • • • • • • • •	4 —		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	.	attrit	outes may be selecte	utes which imply protection. Multiple d in a single sequence. This sequence	
• • • • • • • • • • • • • • • • • • • •	5					does.r	ot chanse any attri	butes of the characters displayed on	
					; •	the so	reen or of the char	acters received. The sequence merely	
	8		• • • • • • • • • • • • • • • • • • • •		•••••	are in	iterpreted by the di	cters with the specified attributes	
	<u>.9</u>								
					;	ENTRY	none		
	.11		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •					
	13 14	· · · · · · · · · · · · · · · ·				EXIT	none		
	15 16				, ,	USES	all		
	16 17								
• • • • • • • • • • • • • • • • • • • •		, 5 5	F0		ASPF:	MÓV	B, #0		
	19 20 0A68			00	HOLLI.	MOV	B, #V	; Initialize byte	
	20 0A68	31	Di	••••••	ASPF1:	ACALL	ĞNP	; Get next parameter	
• • • • • • • • • • • • • • • • • • • •	.21 0A6A .22 0A6C	<u>40</u>	22			JC CJNE	ASPF2	; Exit when done	
	23 OA6F	75	FO	00		MOV	A, #0, ASPF1A B, #0	; None protected	
	24 0A72	B4	02	02	ASPF1A:	CUNE	A, #2, ASPF1B	, none protected	•••••
• • • • • • • • • • • • • • • • • • • •	25 0A75 26 0A77	<u>D2</u>	F3	öż	ASPF18:	SETB	B.3	; Half intensity	
	27 OA7A	D2	F1	02	HOFFID:	CUNE SETB	A, #4, ASPF1C B.1	; Under line	
	28 0A7C	B4	ö5	02	ASPF1C:	CUNE	A, #5, ASPF1D	, onder time	
• • • • • • • • • • • • • • • • • • • •	29 0A7F 30 0A81	<u>D2</u> 	F2	öż	ASPF1D:	SETB	B.2	; Blink	
	31 0A84	D2	F0	02	HOLLID:	CUNE SETB	A, #7, ASPF1E B.O	; Reverse video	
	32 0A86	B4	FÉ	93	MASPFIE:	CUNE	A, #254, ASPF1F	, weverse video	• • • • • • • • • • • • • • • • • • • •
• • • • • • • • • • • • • • • • • • • •	.33 0A89 .34 0A86		Fo	80	ASPF1F:	MOV	B, #1000000B	; No attributes are protected	
	35		r.u		HOFFIF:	"SUMP	ASPF1		
***********	36 0A8E 37 0A91	85	Fo	3F	ASPF2:	MÖV RET	PFIELD, B	; Final change	••••••
•••••			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		

••••••	• • • • • • • • • • • • • • • • • • • •	•••••	• • • • • • • • • • • • • • • • • • • •						
									•••••
••••••			• • • • • • • • • • • • • •						
~~	• • • • • • • • • • • • • • • • • • • •								
S									
			• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		
• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		
·····								•••••	
			• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		•••••••••••••••••••••••••••••••••••••••	
***********	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •							

	1				; ;	CBA -	convert byte to AS	BCII	
	ž Ž	• • • • • • • • • • • • • • • • • • • •							
3	3 4					*UBA*	converts a pyte 1	nto its ASCII equivalent.	
						ENTRY	(A) = byte to co	nnvert	
7	5 7 				, , , , , , , , , , , , , , , , , , ,				
8	 9				5	EXIT	(B) = hundreds ((DPH) = tens dis		
i							(DPH) = tens dis (DPL) = ones dis	git	
	<u>1</u> 2					USES	A, B, DPTR		
	3								
15	5 0A92	75 84	FO	64	CBA:	MOV	B, #100 AB		
16	6 0A95 7 0A96	84 CO				DIV	AB ACC	; Save hundreds amount	:
18	8 0A98	E5	E0 F0			MOV	ACC A, B		
	9 0A9A 0 0A9D	75 	F0	<u>oa</u>		MOV	B, #10 AB	; (A) = tens (B) = or	 1es
2:	1 0A9E	24 	30 83			ADD	A, #′O′ DPH, A	; Make tens disit ASCI ; Save it	I I
	2 0AA0 3 0AA2	F5 E5	83 F0			MOV MOV	A, B		
24	4 OAA4	E5	F0			ADD MOV	Α, #'0' DPL, Λ	; Make ones digit ASC) ; Save it	II
$\frac{^{25}}{26}$	5 0AA6 6 0AA8	<u>F5</u>	82 E0			POP	ACC		
	7 0000	24 F5	30 F0			ADD	A, #101	: Make bundreds digit	ASULL
	O OAAC	CE	En			MITIU	В, А	; Save it	
28	8 OAAC	F5 22	FO			MOV RET	В, А	; Make hundreds digit. ; Save it	
28	8 OAAC 9 OAAE	F5 22	F0			MOV	В, А	; Save it	
28	8 OAAC	F5 22	FO			MOV	В, А	; Save it	
28	8 OAAC	F5 22	FO			MOV	В, А	; Save it	
28	8 OAAC	F5 .22	FO			MOV	В, А	; Save it	
28	8 OAAC	F5 . 22	FO			MOV	В, А	; Save it	
28	8 OAAC	F5 22	FO			MOV	В, А	; Save it	
28	8 OAAC	F5 .22	FO			MOV	В, А	; Save it	
28	8 OAAC	F5 22	FO			MOV	В, А	; Save it	
28	8 OAAC	F5 22	FO			MOV	В, А	; Save it	
28	8 OAAC	F5 22	FO			MOV	В, А	; Save it	
28	8 OAAC	F522	FO			MOV	В, А	; Save it	
28	8 OAAC	F5 22	FO			MOV	В, А	; Save it	
28	8 OAAC	F5 22	FO			MOV	В, А	; Save it	
28	8 OAAC	F5 22	FO			MOV	B, A	; Save it	
28	8 OAAC	F5 22	FO			MOV	В, А	; Save it	
28	8 OAAC	F5 22	FO			MOV	В, А	; Save it	
28	8 OAAC	F5 22	FO			MOV	В, А	; Save it	

ACPR	1							
4			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • •		ACPR 	.ANSI cursor posi	tion report
1	<u>ş</u>					*ACPR*	.outruts.the.curr	ent cursor position in the ANSI form
S	4				;	of ES	C[P];PcR	Please note the tricky code to save a
2	6	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •]	ተቋ₩ኦ.አ	tes.of.code	
1					;			
10	8				,	ENTRY	none	
11								
12					;	EXIT	none	
13	12	• • • • • • • • • • • • • • • • • • • •				HSES	all	
15, OARF 31					•	0.000	211	
16 OAB1 B4 O6 S5 C.NE A. #65 APR 1 IF not 6 THEN return 17.0APR 74 18 MOV A. #ESC 1 Output an ESC 18.0AB6 S1 CC ACALL ACPS 1 Output an ESC 20.0AB8 S1 CC ACALL ACPS 1 Output an ESC 20.0AB8 S1 CC ACALL ACPS 1 Output a fc 20.0AB8 S1 CC ACALL ACPS 1 Output a fc 20.0AB8 S1 CC ACALL ACPS 1 Output a fc 20.0AB8 S1 CC ACALL ACPS 1 Output a fc 20.0AB8 S1 CC ACALL ACPS 1 Output a fc 20.0AB8 S1 CC ACALL ACPS 1 Output a fc 20.0AB8 S1 CC ACALL ACPS 1 Output a fc 20.0AB8 S1 CC ACALL ACPS 1 Output a fc 20.0AB8 S1 CC ACALL ACPS 1 Output a fc 20.0AB8 S1 CC ACALL ACPS 1 Output a fc 20.0AB8 S1 CF ACALL ACPS 1 Output a fc 20.0AB8 S1 CF ACALL ACPS 1 Output a fc 20.0AB8 S1 CF ACALL ACPS 1 Output a fc 20.0AB8 S1 CF ACALL ACPS 1 Output a fc 20.0AB8 S1 CF ACALL ACPS 1 Output a fc 20.0AB8 S1 CF ACALL ACPS 1 Output a fc 20.0AB8 S1 CF ACALL ACPS 1 Output a fc 20.0AB8 S1 CF ACALL ACPS 1 Output a fc 20.0AB8 S1 CF ACALL ACPS 1 Output a fc 20.0AB8 S1 CF ACALL ACPS 1 Output a fc 20.0AB8 S1 CF ACALL ACPS 1 Output a fc 20.0AB8 S1 CF ACALL ACPS 1 Output a fc 20.0AB8 S1 CC ACPS								
17 ORB64 57			<u>D</u> 1	· · · · · <u>· · · ·</u> · · · · ·	ACPR:			; Get parameter
18 OABS 51 CC				35				; IF not 6 THEN return
20				• • • • • • • • • • • • • • • • • • • •				
20 0ABA 51 CC ACAL ACP3 ; Outrut a '[' 22 0ABE 04] 21 0ABE 55 12 MOV A1 YPOS ; Outrut a '[' 22 0ABE 04] 22 0ABE 04 INC A1 YPOS ; Bump to normalize 23 0ABF 51 CF ACAL ACP1 ; Outrut Y-resition 24 0AC1 74 3B CC ACAL ACP3 ; Outrut secretor 25 0AC3 51 CC ACAL ACP3 ; Outrut secretor 26 0AC5 E5 11 MOV A XFOS ; Bump to normalize 28 0AC8 51 CF ACAL ACP1 ; Outrut X-resition 29 0ACA 74 52 ACAL ACP1 ; Outrut X-resition 30 0AC6 02 022A ACP3: LUMP PCOP ; Outrut 'R' and return 31 0ACC 02 022A ACP3: LUMP PCOP ; Outrut 'R' and return 32 0ACF 51 92 ACP1: ACAL CBA ; Convert position to ASCII 34 0ADI AD 92 MOV TEMP, DPL ; Save one's disit 35 0ADI SE 93 MOV TEMP, DPL ; Save one's disit 36 0AD5 B4 31 00 ACP1A: CUMP A BT1 ACP1A ; Outrut tem's disit 37 0ADB 40 02 ACP1A: ACP1A ACP2 ; If not zero THEN 38 0ADA 51 CC ACP2: ACP1A: ACP1A ; Outrut tem's disit 40 0ADD 80 ED SUMP ACP2 ; OUTPUT one's disit and return								; Output an ESC
1			CC	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			
22 OABE 04	21.0ABC							; Output a 'L'
23 OACE 51 CF ACALL ACP1 : Output Y-resition 24 OACE 74 3B MOV A # */ 1. Output Y-resition 25 OACS 51 CC ACALL ACP3 : Output seperator 26 OACE 55 11 MOV A XPOS : OACE 55 12 CF ACALL ACP3 : OACE 55 12 CF ACALL CBA I CONVERT POSITION 50 ACCID 33 OACE 51 S2 ACP1: ACALL CBA I CONVERT POSITION 50 ACCID 35 OADS 55 S3 MOV TEMP, DPL I SAVE ONe'S disit 55 OADS 55 S4 31 OO CUNE A, # */ ACP1 I GET Len's disit 57 OADS 40 O2 ACP14: JC ACP2 I IF NOT ZEPO THEN 38 OADS 51 CC ACCAL ACP3 I OUTPUT Len's disit 1 OACE 50 OADS 55 OADS 55 CF ACALL ACP3 I OUTPUT Len's disit 1 OACE 50 OADS 55 OADS 55 OADS 55 CF ACALL ACP3 I OUTPUT Len's disit 1 OACE 50 OADS 55 OA		04			• • • • • • • • • • • • • • • • • • • •			: Rump to populito
24 OAC1			ÇF					; Output Y-position
26 OACS ES 11						MOV	A, #′;′	·····
27.06C7			ÇÇ					; Output seperator
28 0ACS 51 CF MOV A #/R' 1 Output X-position 30 0ACC 02 022A ACP3: LJMP PCOF I Output (R' and return 31 0ACC 02 022A ACP1: ACALL CBA I CONVERT Position to ASCII 32 0ACF 51 92 ACP1: ACALL CBA I CONVERT POSITION to ASCII 33 0ACF 51 92 ACP1: ACALL CBA I CONVERT POSITION to ASCII 34 0AD1 AD 82 MOV TEMP, DPL I SAVE ONE'S digit 35 0AD3 E5 83 MOV A. DPH I Get ten's digit 37 0AD5 B4 31 00 CJME A #/I', ACP1A 37 0AD5 B4 0. 02 ACP1A: JC ACP2 I IF not zero THEN 38 0ADA 51 CC ACALL ACP3 39 0ADC ED ACP2: MOV A, TEMP I output ten's digit and return 40 0ADD 80 ED SJMP ACP3 ; Output one's digit and return								
22, OACA 74 52 MOV A #/R' , Softat X-Puillon 30 31, OACC 02 022A ACP3: LJMP PCOF i Qutrut /R' and return 32 33, OACF 51 92 ACP1: ACALL CBA j Convert Position to ASCII 34 0AD1 AD 82 MOV TEMP, DPL ; Save one's digit 35, OAD3 E5 83 MOV A, DPH ; Get tem's digit 36 0AD5 B4 31 00 CJNE A #1', ACP1A ; Get tem's digit 37, OAD8 40 02 ACP1A: JC ACP2 ; IF not zero THEN 38 0ADA 51 CC ACP1A: JC ACP3 ; output tem's digit 40 0ADD 80 ED ACP2: MOV A, TEMP ; output one's digit and return			CF	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			; Bump to normalize
30 31.0ACC 02 022A ACP3: LUMP PCOF : Qutput (R' and return 32.0ACF 51 92 ACP1: ACALL CBA : Convert position to ASCII 34 0AD1 AD 82 MOV TEMP, DPL : Save one's disit 35.0AD3 E5 83 MOV A: DPH : Get ten's disit 37.0AD5 B4 31 00 CINE A: **1', ACP1A : Get ten's disit 37.0AD5 A00 02 ACP1A: UC ACP2 : IF not zero THEN 38.0AD4 51 CC ACP3 : output ten's disit 40.0AD5 80 ED SUMP ACP3 : Output one's disit and return 39.0ADC ED SUMP ACP3 : Output one's disit and ret								; Uutput X-position
32 33 OACF 51 92 ACP1: ACALL CBA 1 Convert Position to ASCII 34 OAD1 AD 92 MOV TEMP, DPL ; Save one's digit 35 OAD5 B4 31 00 CJNE A, #1', ACP1A 37 OAD8 40 02 ACP1A: JC ACP2 ; IF not zero THEN 38 OADA 51 CC ACP2: MOV A, TEMP 40 OADD 80 ED SJMP ACP3 ; Output one's digit and return	30			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			
32 OACF 51 92 ACP1: ACALL CBA : Convert position to ASCII 34 OAD1 AD 82 MOV TEMP, DPL : Save one's digit 35 OAD3 E5 83 MOV A. DPH : Get ten's digit 36 OAD5 B4 31 00 CJNE A. #'1', ACP1A 37 OAD8 40 02 ACP1A: JC ACP2 : IF not zero THEN 38 OADA 51 CC ACP2: MOV A. TEMP : output ten's digit 40 OADD 80 ED SJMP ACP3 : Output one's digit and return		92	022A		ACP3:	LUMP	PCOF	; Output 'R' and return
34 OAD1 AD 82 MOV TEMP, DPL ; Save one's digit 35 OAD5 B4 31 OO CJNE A, #1', ACP1A 37, OAD8 40 02 ACP1A: JC ACP2 ; IF not zero THEN 38 OAD5 51 CC ACALL ACP3 ; output ten's digit 40 OADD 80 ED SJMP ACP3 ; Output one's digit and return		ET 4						
35.0AD3 E5 83 MOV A, DPH Save one's disit 36.0AD5 B4 31 OO CJNE A, #/1', ACP1A Get ten's disit 37.0AD8 40 O2 ACP1A:			····. 25. · · · ·		ACP1:			
36 OADS								
37.0ADS 40 02 ACP1A: UC ACP2 ; IF not zero THEN 38.0ADA 51 CC ACP2: ACALL ACP3 ; output ten's digit 39.0ADC ED ACP2: MQV A, TEMP 40.0ADD 80 ED SJMP ACP3 ; Output one's digit and return		B4	31					; Get ten's digit
39 OADC ED ACP2: MOV A. TEMP , OUTPUT ten's digit and return 40 OADD 80 ED SJMP ACP3 ; Output one's digit and return		40		••	ACP1A:			* TE set see TUCK
39 OADC ED ACP2: MOV A TEMP 40 OADD 80 ED SJMP ACP3 ; Output one's digit and return		51	cc	•••••		ACALL		
SJMP ACP3 ; Output one's digit and return	39.0ADC				ACP2:	MOV		, odthat ten 5 dibit
	40 OADD	80	ED			SJMP	ACP3	; Output one's digit and return
			•••••••					, Ducput one's digit and return
			•••••	•••••		• • • • • • • • • • • • • • • • • • • •	•••••	•••••••
			•••••	• • • • • • • • • • • • • • • • • • • •				
							• • • • • • • • • • • • • • • • • • • •	***************************************

	1					APRNT -	ANSI print page		
	2				; ;	*APRNT*	prints the page to	the opposite port.	
4	7								
	5				5	ENTRY	none		
						EXIT	none		• • • • • • • • • • • • • • • • • • • •
	9					USES	all		
10 11 11) 1				; 	USES	a		
1:	2 3 OADF	11	2A		APRNT:	ACALL	SETDISP	; Set dispatch and return	
14	4								
$\frac{15}{16}$	5 OAE1 6 OAE3	<u>11</u>	2F 37			ACALL	SETNORM A, #171, APR	; Restore dispatch address ; IF ESC # 7 THEN	
1	7 OAE6 3 OAE9	02 22	103C		APR:	JMP RET	R2_PRNT	; print page	
13 13 20	3 OAE9 9				HFR•				
20 22.	0								
2:	2					ACDD	ANSI set baud rate.		
23 24	4			• • • • • • • • • • • • •	· · · · ? · ? · · · · · · · · · · · · ·				
2! 24	5					*ASBR*.	sets the baud rate.	to that specified by Pn.	,
2	7								
2: 2:	8 ?				; ;	ENTRY	none		
3,	0				5	EXIT	none		
3:	1 2					USES	all		
:۾ ع	3								
3	5QAEA	31	p1		ASBR:	AÇALL JNC	GNP	<pre>5 Get parameter 7 IF no parameter THEN</pre>	
3:	6 OAEC 7 OAEE	50 04	01			INC	^	; default to 1	
3	8 OAEF	B4	OE	00	ASBR1:	CUNE UNÇ	A, #14, ASBR2 AFR	; IF in range (0,.13).THEN.	
	9QAF2 o oAF4	59 C1	F.5 4B		ASBR2:	AJMP	SBR	; set baud rate and return	
		. .							

						• • • • • • • • • • • • • • • • • • • •		
OM; MICRO Z-29 COMP	SENCH BOS	51 CROSS ASSE MINAL FIRMWAR	MBLER (V1)-218 2 E VERSION 1,03 **	24-FEB-83 15 **	:30:06 PAGE 96			
		•••••		ARAMP	- ANSI reset all) modes to power up con	figuration	
3 4			; ;			s.to.power.up.configura		
5 6 				*ARAMP	* and *RAMP* res er up.	sets all flags etc. to	the configuration	
9 0 11	· • • • • • • • • • • • • • • • • • • •		7 9 7	ENTRY	none		••••••	
.11 12 13				EXIT	none	•••••		
13 14 15			······································	USES	all	•••••••••••	••••••	
16 17 OAF6 18 OAF8 19 OAFA	31 50 02	D1 EF	ARAMP:	ACALL JNC	GNP APR	; Get any par ; IF no param		
	92	1003	RAMP:	JMP	R2_IN2	; reset and	i return	
			•••••••				·····	
				•••••				
		•••••		•••••				

Z29ROM; MICROBENCH 8051 CROSS AS *** Z-29 COMPUTER TERMINAL FIRM	SSEMBLER (V1)-218 24-F NARE VERSION 1.03 ***	EB-83 15:	30:06 PAGE 97
1	; ;	WEOL -	wrap at end of line
2	*	*WEOL*	sets the flas which causes the terminal to perform lase return and line feed immediatly after the 80th
4	**************************************	a carri	age return and line feed immediatly after the 80th ter before a line is displayed.
56		Charact	er berore a time is discissed.
7	.	ENTRY	none
8 9	······································	EXIT	
10 11	; ;		none
12 13	\$	USES	none
14			AUTOURAD . Flag auto wran
15 OAFD D2 2B 16 OAFF 22	WEOL:	SETB	AUTOWRAP ; Flas auto wrap
17			
18 19			
20 21	; ;	DEOL -	discard at end of line
22	;		
23	;	charact	resets the wrap around at end of line flas. ters past column 79 are placed in column 79 until
25 26		a.sann	iase return is received.
27		ENTRY	none
28 29	; ;		
30 31	;	EXIT	none
32	; ;	USES	none
35_0B00C22B	DEOL:	ÇLR RET	AUTOWRAP : Clear auto wrap
36 0B02 22			
,			

)

)

)

)

)

Z29ROM *** Z-	1; MICROBENCH 80 29 COMPUTER TER	951 CROSS ASSEMBL RMINAL FIRMWARE V	ER (V1)-218 2	24-FEB-83 15:30:06 PAGE 98
	<u>1</u> 2	•	;;	EAM - enter ANSI mode
	3 4		; ;	*EAM* places the terminal in the ANSI mode for escape codes.
	5 6 7		,	ENTRY none
	8 9		***************************************	EXIT none
	0		**************************************	USES none
1:	3 OBO3 75 4 OBO6 22 5	28 01	EAM:	MOV MODE, #00000001B ; Place in ANSI mode RET
	8			
		•••••	5.7	EZM — enter ZDS mode
21 22 23	1 2 2	• • • • • • • • • • • • • • • • • • • •	····	*EZM* takes the terminal out of ANSI mode and into ZDS mode.
24	4	••••••••	· · · · · · · · · · · · · · · · · · ·	ENTRY none
26 27 28	5 6 7	•••••	· · · · · · · · · · · · · · · · · · ·	EXIT none
28 29	8 9 0		***************************************	USES none
			. 	
30 31 32	0 1 0B07 75 2 0B0A 22	28 02	EZM: RT1:	MOV MODE, #0000010B ; Take out of ANSI mode
30 31 32	0 1 0807 75 2 080A 22	28 02	EZM: RT1:	MOV MODE, #00000010B ; Take out of ANSI mode RET
30 31 32	0 1 0B07 75 2 0B0A 22	28 02	EZM: RTI:	MQV MODE, #00000010B ; Take out of ANSI mode RET
30 31 32	0 1 0B07 75 2 0B0A 22	28 02	EZM: RT1:	MOV MODE, #00300010B ; Take out of ANSI mode RET
30 31 32	0 1 0B07 75 2 0B0A 22	28 02	EZM: RT1:	MOV MODE, #0000010B ; Take out of ANSI mode RET
30	0 1 0B07 75 2 0B0A 22	28 02	EZM: RT1:	MOV MODE, #0000010B ; Take out of ANSI mode RET
30 31 32	0 1 0B07 75 2 0B0A 22	28 02	EZM: RŤi:	MOV MODE, #0000010B ; Take out of ANSI mode RET
30 31 32	0 1 0B07 75 2 0B0A 22	28 02	EZM: RT1:	MOV MODE, #0000010B ; Take out of ANSI mode RET
30 31 32	0 1 0B07 75 2 0B0A 22		EZM: RT1:	MOV MODE, #0000010B ; Take out of ANSI mode RET
30 31 32	0 1 0B07 75 2 0B0A 22	28 02		MOV MODE, #0000010B ; Take out of ANSI mode RET
31	1 OBO7 75 2 OBOA 22			
31	1 OBO7 75 2 OBOA 22			
31	1 OBO7 75 2 OBOA 22			

C

				; ;	E25L -	enable 25th line	
				5			ov of the 25th line only if it has
					not be	en previously enab	ay of the 25th line only if it has led. The 25th line is cleared at
i						me it is enabled.	
,				7 9			
, 					ENTRY	none	
, ,				.	·····EXIT	none	
				,	USES	A, B, DPTR, TEMP	FIRI
		4=			JB	L25EN, RT1	• IC applied THEN pature
OBOB OBOE	20 02	<u>15</u>	FC	E25L:	·····ŠĒTB···	L25EN,	; IF enabled THEN return ; ELSE enable 25th line
OB10	75 61	F0	18		MOV	B, #MAXLINE	
OBIS	61	96			AJMP	CLRLINE	; Clear 25th line
, , ,							
· · · · · · · · · · · · · · · · · · ·							
3					D25L -	disable 25th line	
; ; 				;	*D25L*	disables the disp	lay of the 25th line and clears it.
·······				; -	If the	cursor is on the	25th line then it is sent to the
' }				.	home.r	osition.	
)					ENTRY		
) 				; ;		none	
2				***************************************	EXIT	none	
3					USES	A, B, DPTR, TEMP	, PTR1
, 7. OB15	C2	15		D25L:	CLR	L25EN	; Disable 25th line
OB17	C2 75	15 F0	18		MOV	L25EN B, #MAXLINE	
OB1A	71 E5	96 12			ACALL	CLRLINE	; Clear 25th line
OB1C OB1E	E5 B4	12 18	02		MOV CJNE	A, YPOS A, #MAXLINE, XR2	DISP : IF cursor on 25th line THEN
OB21	91	36	×÷		ACALL	SCH	_DISP
3 - OB23	02	1042		XR2_DISP:	LJMP	R2_DISP	; Display any status

Z29R0M;	MICROB	ENCH 805	1 CROSS ASSI	EMBLER (V1)-218 2	24-FEB-83 15	:30:06 PAGE 100	······································
trr.Atr	47. 99009 1	15K15KM		RE_VERSION_1.03 **		enable kerboard	click
	2 3 4 5			; , ; ;	*EKC* /	resets the flas s the command to	that disables keyboard click and the keyboard.
			• • • • • • • • • • • • • • • • • • • •	**************************************	ENTRY	none	
, 10	- 9 5			; ;	EXIT	none	
11 12 13			• • • • • • • • • • • • • • • • • • • •		USES	all	
14 15	1 0826 5 0828	D2 74	33 82 026D	EKC:	SETB MOV	CLKF A, #KBC_EC	; Turn on key click
16 17 18	5 082A 7 3	02	026D	XPCKB:	JMP	PCKB	; Output command and return
	» 		• • • • • • • • • • • • • • • • • • • •				
21 22 23 23		•••••		5 5 5		o keyboard clic	k at disables keyboard click and
23 24 25 28	j 5			; ;	outputs	s the command to	the keyboard.
27 28	» 7 3				ENTRY	none	
29 30)				EXIT	none	
31 32 33	}				USES	.all	
35	0825 082F 0831	74 80	33 83 F7	NKC:	CLR MOV SJMP	CLKF A, #KBC_DC	; Turn off key click
					SUMP	XPCKB	; Output command and return
		• • • • • • • • • • • • • • • • • • • •					
· · · · · · · · · · · · · · · · · · ·	• • • • • • • • • • • • • • • • • • • •		•••••				
			••••••				
v							
				*			
***************************************			••••••				
•••••			• • • • • • • • • • • • • • • • • • • •			·····	

7-29 COMPU	ITER TERM	INAL FIRMWARE	VERSION 1.03 **	4-FEB-83 15: *	30:06 PAGE 101	
1			; ;		set block cursor	
2 3			* *	*SBC* i	rograms the CRTC t	o display a block cursor.
4 5			* *	······PNYAU··		
6 5			; ;	ENTRY	none	
9 10			,	USES	none ''A, 'B, 'DPTR'	
10 11 12						·
13 0B33 14 0B35	80	32 02	SBC:	CLR SUMP	CRUL "SUC1"	; Clear underline flas ; Initialize cursor format
15 16 17						
17 18 19 20			5 5	suc –	set underscore curs	or
20 21 22				*SUC*	programs the CRTC t	o display an underline cursor.
23 24				ENTRY	none	
25 26			· · · · · · · · · · · · · · · · · · ·	EXIT	none	
27 28 29				USES	A, B, DPTR	
29 30 31 0B37 32	D2	32	suc:	SETB	CRUL	; Set underline flas
32 33 OB39	02	101B	SUC1:	JMP	R2_ICUR	; Initialize cursor format
		,				

			******************	RSION 1.03 ***				
	.1			5 . 5	EHSM -	enter hold screen	node	
				; ;		sets the hold scr		lan.
***************************************	4 5			····	THURUS	. 54.53 tile . 110.10 \$.51°	een inione 't	145.
••••••	6		• • • • • • • • • • • • • • • • • • • •	"	ENTRY	none		
••••••	<u>7</u> 8	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		EXIT	none		
	9 10	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		USES	none		
	11 12							
	13 OB3C 14 OB3E	D2 75	36 1E 01	EHSM:	SETB	HSMODEF		Enter mode Set number of lines
	15 OB41 16	73 80	02		MOV SJMP	HSLINE, #1 XHSM1	; ;	Set number of lines Clear stop display and return
	16 17					••••••••••	• • • • • • • • • • • • • • • • • • • •	·····
	17 18 19		• • • • • • • • • • • • • • • • • • • •			•••••••		
	19 20		• • • • • • • • • • • • • • • • • • • •	**************************************	XHSM -	exit hold screen	mode	
· · · · · · · · · · · · · · · · · · ·	21 22		• • • • • • • • • • • • • • • • • • • •		*XHSM*	clear hold screen	mode flas	•
······	23 24							
	25 26				ENTRY	none		· · · · · · · · · · · · · · · · · · · ·
	27 28				EXIT	none		
	29 30			; ;	USES	none		
3	30 31 32 0843							
3	33	C2	36	XHSM:	CLR	HSMODEF	5	Exit mode
	34 OB45 35 OB47	Ć2 22	22	XHSM1:	CLR	HSSTOPF	· · · · · · · · · · · · · · · · · · ·	Clear stop display flas
• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	RET		• • • • • • • • • • • • • • • • • • • •	•••••
				• • • • • • • • • • • • • • • • • • • •				
• • • • • • • • • • • • • • • • • • • •								
							••••••	
					•••••			

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER *** Z-29 COMPUTER TERMINAL FIRMWARE VERS	(V1)-218 24-FEF	9-83 15 :	30:06 PAGE 103
		EKSM -	enter keypad shifted mode
2 3	፣ ፤	*EKSM*	places the numeric keypad into the shifted mode.
4 5	; ;		
<u> </u>	事 . [‡]	ENTRY	none
8 9 10	ቹ . ፣	EXIT	none
10 11 12	;	USES	none
12 13 OB48 D2 30	EKSM:	SETB	KPADSHTF ; Set keypad shifted
14 OB4A 22		RET	
15 16 17			
17 18 19		xksm -	exit_keypad_shifted_mode
19 20 21	; ;		takes the numeric keyead out of the shifted mode.
	* •	. 20004000	APOPA . AVA. // FUT/. TA . 111.0. AT . 101.0. 10. FUT . TO FO FO FO FO
24		ENTRY	none
25 26		EXIT	none
27. 28	.? \$	USES	none
29 30			VENEROUTE . Class bound shifted
31 0848 C2 30 32 084D 22	.XK\$M:	RET	KPADSHTF ; Clear keypad shifted
		.	

Z29ROM; MICROBENCH 8051 CROSS ASS *** Z-29 COMPUTER TERMINAL FIRMWA	SEMBLER (V1)-218 2 ARE VERSION 1.03 **	4-FEB-83 15:30:06 PAGE 104 *	
1		EKAM – enter keypad alteri	nate mode
2	; 	*EKAM* places the numeric	keypad into the alternate mode.
5	, , ,	ENTRY none	
7 8	; ;	EXIT none	
9 10 11		USES none	
12 13 084E D2 31 14 0850 22	EKAM:	SETB KPADALTF RET	; Set keypad alternate
15 16 17			
18 19		XKAM — exit keypad alterna	ate mode
20 21 22			errad out of the alternate mode.
23 24	, , , , , , , , , , , , , , , , , , ,	ENTRY none	
25 26 27		EXIT none	
29 30	**************************************	USES none	
30 31 OB51 C2 31 32 OB53 22	XKAM:	CLR KPADALTF RÉT	; Clear keypad alternate
	XKAM:	CLR KPADALTF RÉT	; Clear keypad alternate
	XKAM:	CLR KPADALTF RÉT	; Clear keypad alternate
	XKAM:	CLR KPADALTF RÉT	; Clear keypad alternate
	XKAM:	CLR KPADALTF RÉT	; Clear keypad alternate
	XKAM:	CLR KPADALTF RET	; Clear keypad alternate
	XKAM:	CLR KPADALTF RET	; Clear keypad alternate
	XKAM:	CLR KPADALTF RET	; Clear keypad alternate
	XKAM:		
31 0B51 C2 31 32 0B53 22			
31 0B51 C2 31 32 0B53 22			
31 0B51 C2 31 32 0B53 22			

Z29ROM; MICROBENCH 8051 CROSS ASSE *** Z-29 COMPUTER TERMINAL FIRMWAF	ÉMBLÉR (V1)-218 24	-FEB-83 15:30:06 PA	GE 105
1	;;		sert character mode
2 3	; ;	••••	erminal into insert character mode.
5	ş ,		
7 8	\$ \$	ENTRY none EXIT none	
9 10		EXIT none USES all	
11	,		
i2 13 OB54 D2 21 14 OB56 61 23	EICM:	SETB ICMODEF AJMP XR2_DIS	; Set insert mode flag ; Display any change in status
15 16 17 18			
18 19 20		XICM - exit ins	sert character mode
21 22	; ;	*XICM* takes te	erminal out of the insert character mode.
23 24		ENTRY none	
25 26	 ; :	ĖXIT none	
27 28 29		USES all	
29 30 31 0858 C2 21 32 085A 61 23	XICM:	CLR ICMODEF AJMP XR2_DIS	
	XICM:	CLR ICMODEF AJMP XR2_DIS	: Clear insert mode flas SP ; Diselay any chanse in status
	XICM:	CLR ICMODEF AJMP XR2LDIS	; Clear insert mode flas SP ; Display any change in status
	XICM:	CLR ICMODEF AJMP XR2_DIS	: Clear insert mode flag 3P : Display any change in status
	XICM:	CLR ICMODEF AJMP XR2_DIS	; Clear insert mode flas SP ; Display any change in status
	XICM:	CLR ICMODEF AJMP XR2_DIS	; Clear insert mode flag
	XICM:	CLR ICMODEF ÄJMP XR2_DIS	Clear insert mode flag Display any change in status
	XICM:	CLR ICMODEF AJMP XR2_DIS	Clear insert mode flas Display any change in status
	XICM:	CLR ICMODEF AJMP XR2_DIS	Clear insert mode flas Display any change in status
31 OB58 C2 21 32 OB5A 61 23		AJMP XR2_DIS	
31 OB58 C2 21 32 OB5A 61 23		AJMP XR2_DIS	
31 OB58 C2 21 32 OB5A 61 23		AJMP XR2_DIS	
31 OB58 C2 21 32 OB5A 61 23		AJMP XR2_DIS	

))

)

)

)

)

•

*** Z-	M; MICROBENC -29 COMPUTER	H 8051 CROSS ASSEN TERMINAL FIRMWARE	1BLER (V1)-218 VERSION 1.03 *	24-FEB-83 15:30:06 PAGE 106 **	
	.1			EC - enable cursor	
	.3 		.	*EC* enables the displa	y of the cursor.
	. 5		• • • • • • • • • • • • • • • • • • •	······	
	.7 8			ENTRY none EXIT none	
	.9 10		* · · · · · · · · · · · · · · · · · · ·		
	11 12		,	USES none	
	13 OB5C D1 14 OB5E 22	21A	EC:	SETB ENBLOUR RET	; Enable display of cursor
	15 16	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		
	17 18	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		
	19 20	•••••		DC - disable cursor	
	21 22 22			*DC* disables the displa	ay of the cursor
2	23 2 4			ENTRY none	
2	25 26			EXIT none	
2	2 7 28		····	USES none	
3	29 30				,
3	31 OB5F C2 32 OB61 22	1A	DC:	CLR ENBLCUR RET	; Disable display of cursor
3 3	31 OB5F C2 32 OB61 22	1A	DC:	CLR ENBLCUR RÉT	; Disable display of cursor
3	31 0B5F C2 32 0B61 22	1A	DC:	CLR ENBLCUR RET	; Disable display of cursor
3	31 0B5F C2 32 0B61 22	1A	DC:	CLR ENBLCUR RÉT	; Disable display of cursor
3	81 0B5F C2 82 0B61 22	1A	DC:	CLR ENBLCUR RET	; Disable display of cursor
	31 0B5F C2 32 0B61 22	1A	DC:	CLR ENBLCUR RET	; Disable display of cursor
	31 0B5F C2 32 0B61 22	1A	DC:	CLR ENBLCUR RET	; Disable display of cursor
	31 0B5F C2	1A	DC:	CLR ENBLCUR RET	; Disable display of cursor
	31 0B5F C2	1A	DC:	CLR ENBLCUR RET	; Disable display of cursor
3	31 0B5F C2	1A	DC:	CLR ENBLCUR RET	; Disable display of cursor
	31 0B5F C2	1A	DC:	CLR ENBLCUR RET	; Disable display of cursor
	81 OBSF C2			RET	
	31 OBSF C2			RET	
	31 OBSF C2 32 OB&1 22			RET	
	31 OBSF C2 32 OB&1 22			RET	

C C C C C C C C C C

Z29ROM;	MICROBENCH	8051 CROSS ASSE	MBLER (V1)-218 2 E VERSION 1.03 **	4-FEB-83 15:	:30:06 PAGE 107		
	·				enable auto carr	iase return	
2 3 4	2		5 5 7 8	*EACR*	enables the term upon receipt of	inal to perform an automatic c a line feed character.	arriase
6 7	? > ?		\$ \$	ENTRY	none		
		• • • • • • • • • • • • • • • • • • • •	,		none		
10 11		.,		USES	none	,	
14	2 3 4 OB62 D2 5 OB64 22	20	EACR:	SETB RET	AUTOCR	; Enter auto CR mode	
1 6 1.7	5 7				,		
18 19	?						
20 2 <u>1</u>	l		;; <u>†</u>		exit auto carria		
22	3			***************************************			
24 25 26	5			ENTRY	none		
27	7			EXIT	none		
29	?			USES	n.qn.e		
32	l 2 OB65 C2	2C	XACR:	CLR	AUTOCR	; Exit auto CR mode	
33	30B6722.			RET			
		• • • • • • • • • • • • • • • • • • • •					
				• • • • • • • • • • • • • • • • • • • •			
		• • • • • • • • • • • • • • • • • • • •					
,							

Z29ROM; MICROBENCH 8051 CROSS ASSEMB *** Z-29.COMPUTER.TERMINAL.FIRMWARE.Y	LER (V1)-218 24 VERSION 1.03 ***	FEB-83 15:30:06 PAGE 108	
		EALFenable.auto.line.feed	
2 3 4 5	; ; ;	*EALF* enables the terminal to perform an automatic	
<u> </u>	; 	ENTRY none	
8 9		EXIT none	
10 11 12		USES none	
13 14 0B68 D2 2D 15 0B6A 22 16	EALF:	SETB AUTOLF ; Enter auto LF mode RET	
17 18		·····	
19 20	; ;	XALF — exit auto line feed mode	
21 22 23 24	**************************************	*XALF* clears the auto line feed mode flag.	•••••
24 25 26	; ;	ENTRY none	
26 27 28	; 	EXIT none	
29 30	; ;	USES none	
30 .		***************************************	
31 32 0868 C2 2D	XALF	CLR AUTOLF ; Exit auto LF mode	
31	XALF:		
31 32 0868 C2 2D	XALF:	CLR AUTOLF ; Exit auto LF mode	
31 32 0868 C2 2D	XALF:	CLR AUTOLF ; Exit auto LF mode	
31 32 0868 C2 2D	XÁĽF:	CLR AUTOLF ; Exit auto LF mode	
31 32 0868 C2 2D	XÁĽĖ	CLR AUTOLF ; Exit auto LF mode	
31 32 0868 C2 2D	XALF:	CLR AUTOLF ; Exit auto LF mode	
31 32 0868 C2 2D	XALF:	CLR AUTOLF ; Exit auto LF mode	
31 32 0868 C2 2D	XALF	CLR AUTOLF ; Exit auto LF mode	
31 32 0868 C2 2D 33 086D 22		CLR AUTOLF ; Exit auto LF mode	
31 32 0868 C2 2D 33 086D 22		CLR AUTOLF ; Exit auto LF mode RET	
31 32 086B C2 2D 33 086D 22		CLR AUTOLF ; Exit auto LF mode RET	
31 32 086B C2 2D 33 086D 22		CLR AUTOLF ; Exit auto LF mode RET	

1			;;	EGATM -	- enter suarded area transfer mode
3			\$ \$		e places the terminal in a mode such that all data
			.	is to be	e transmitted in a data stream such as transmit page.
6 				ENTRY	none
8 9 10			, ,	EXIT	попе
				USES	none
13 14 OB6E 15 OB70		25	EGATM:	CLR RET	GATM ; Clear guarded area transfer
16 17 18					
18 19 20					
20 21 22			; ; 		exit suarded area transfer mode places the terminal in a mode such that only
22 23 24			\$ 	unprote	ected data is to be transmitted in a data stream
24 25 26			; 	such as	transmit page.
26 .27 .28				ENTRY	none
28 29 30				EXIT	none
30 31 32			,	USES	none
32 33 34 0871	D2		XGATM:	SETB	GATM ; Set guarded area transfer
35 0B73	22			RET	
	• • • • • • • • • • • • • • • • • • • •				
		••••			
					

			MINAL.FIRM								
	1					EERM	enter eraşur	re mode			
	2 3				;						
***********	4			• • • • • • • • • • • • • • • •	· · · · · · · · · · · · · · · · · · ·	erased	in functions	s anch as eus	a mode sych t ase screen.	uaragara	. 45
•••••	5	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •								
	7					ENTRY	none				
	9				;	EXIT	none				
	10				,					• • • • • • • • • • • • • • • • • • • •	•••••
• • • • • • • • • • • • • • • • • • • •	<u>11</u>				?	USES	.none	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •
	13 14 OB74	C2	1B		EERM:	CLR	ERM		; Clear era		•••••
	15 OB76	22				RET	ERF1		, crear era:	sure mode	
	16 17										
	17 18										
•••••	<u>19</u> 20	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			XERM -	exit erasure	mode			• • • • • • • • • • • • • • • • • • • •
	<u>21</u> 										
	23 24				, , , , , , , , , , , , , , , , , , ,	unprote	cted data is	erminal in a erased in d	node such the	nat only n as erase	
	24 25				;	screen.					
••••••	. 25 26			• • • • • • • • • • • • • • • • • • • •	••••••••••••	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
• • • • • • • • • • • • • • • • • • • •	27 28			• • • • • • • • • • • • • • • • • • • •		ENTRY	.none				
	29 					EXIT	none				
	. 31 . 32				;	USES	none				
	32 33										
•••••	34 0877	D2	iB	•••••	XERM:	SETB	· ERM · · · · · · · · · ·		; Set erasu	e mode	
• • • • • • • • • • • • • • • • • • • •	35 OB79	22				RET					
	· • • • • • • • • • • • • • • • • • • •									• • • • • • • • • • • • • • • • • • • •	
••••••			•••••	• • • • • • • • • • • • • • • • • • • •							•••••
						• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	•••••
			•••••		• • • • • • • • • • • • • • • • • • • •						•••••
······································											
	••••••			• • • • • • • • • • • • • • • • • • • •							
											•••••
											•••••
		•••••	•••••			• • • • • • • • • • • • • • • • • • • •					
•••••											

				PCLK -	program clock	
1 2 3			; ;	*PCLK*	sets the internal cl	ock.
5 			, , , ,	ENTRY	none	<u></u>
7 				EXIT	none	
9 10				USES	A, DPTR	
11						
	1 D1 4 18		PCLK:	ACALL	GNP A, #24, PCL1A	; Get hour parameter
15 OBZE - 5	0 14	00	PCL1A:	CJNE JNC	net o	; IF bad parameter THEN exit
16 OB81 F 17 OB83 3	5 3D			MÖV	THOUR, A	; Get minutes parameter
18 OB85 E	1 D1 4 3C			CJNE	GNP A, #60, PCL1B	
19 OB88 5 20 OB8A F	ioOB 5 3C		PCL1B:	JNC MOV	PCL2 TMIN, A	; IF bad parameter THEN exit
21 OBSC 3	1			ACALL	GNP A, #60, PCL1C	; Get seconds parameter
22 OB8E E 23 OB91 5	14 3C 10 02 15 3B	00	PCL1C:	CJNE JNC	PCL2	; IF bad parameter THEN exit
				MOV	TSEC, A	
24 0893 F 25 0895 2	5 3B		PCL2:	RET		
24 0B93 F 25 0B95 2	5 3B		PCL2:	RET		
24 0893 F 25 0895 2	5 3B 2		PCL2:	RET		
24 0893 F 25 0895 2	5 3B		PCL2:	RET		
24 0893 F 25 0895 2	5 3B		PCL2:	RET		
24 0893 F 25 0895 2	5 3B		PCL2:	RET		
24 0893 F 25 0895 2	5 3B		PCL2:	RET		
24 0B93 F 25 0B95 2	5 3B 2		PCL2:	RET		
24 0893 F 25 0895 2	5 3B		PCL2:	RET		
24 0893 F 25 0895 2	5 3B		PCL2:	RET		
24 0893 F 25 0895 2	5 3B 2		PCL2:	RET		
24 0893 F 25 0895 2	5 3B		PCL2:	RET		

Z29RO *** Z)M; MICROE 2–29 COMPL	SÉNCH 80	51 CROSS ASS	EMBLER (V1)-218 20 RE VERSION 1.03.**	4-FEB-83 15	:30:06 PAGE 112	
	1					Eclear.line.comp	letely
	2 3 4	•••••••		; ;	*CLRLI		nto the entire line indicated and
	5 6 7	• • • • • • • • • • • • • • • • • • • •					
•••••	8					(B) = line number	to erase
************	9 10 11	• • • • • • • • • • • • • • • • • • • •		·····	USES USES	none	DTD1 HODE2 TNDV2
	11 12 13	• • • • • • • • • • • • • • • • • • • •					PTR1, WORK2, INDX2
	13 14 0896 15 0898 16 0898	C0 75 71	3F 00 A0	CLRLINE: O	PÚSH MOV ACALL	PFIELD PFIELD, #0 CLRL	; Save protected fields ; Make everythins unprotected ; Clear line
	17 OB9D 18 OB9F 19	<u>Do</u>	3F		POP RET	PFIELD	; Restore protected fields
	19 20 21 22			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		
	22 23 24			5 5	CLRL -	clear line	
•••••	24 25 26			; ;	*CLRL*	writes spaces into	the entire line indicated.
	27 28		• • • • • • • • • • • • • • • • • • • •		ENTRY	(B) = line number	to erase
	29 30	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	5	EXIT	none	
;	31 32 33			······································	USES	A, B, DPTR, TEMP,	PTR1, WORK2, INDX2
;	34 35 OBAO	E4		CLRL:	CLR	А	
	36 0BA1 37 0BA4	12 74	0503 50		LCALL MOV	CLA A, #NUMCHARS	; Calculate address ; 80 characters
	38 0BA6 39 0BA9	75 02	F0 00 100C		MÓV LJMP	B, #0 R2_FILL	<pre>7 No attributes 7 Fill with spaces and return</pre>
		· · · · · · · · · · · · · · · · · · ·					
	• • • • • • • • • • • • • • • • • • • •					• • • • • • • • • • • • • • • • • • • •	

	• • • • • • • • • • • • • • • • • • • •				• • • • • • • • • • • • • • • • • • • •		
						•••••••••••••••••••••••••••••••••••••••	
	• • • • • • • • • • • • • • • • • • • •				• • • • • • • • • • • • • • • • • • • •		
	• • • • • • • • • • • • • • • • • • • •		••••••••				
							•••••••••••••••••••••••••••••••••••••••
•••••••							•••••••••••••••••••••••••••••••••••••••
			• • • • • • • • • • • • • • • • • • • •	•••••	• • • • • • • • • • • • • • • • • • • •		

· <u>‡</u>					CLRS -	clear screen	
2 3 4 5		••••••		; ;	*CLRS* is on (clears entire display he 25th line then jus	except the 25th line. IF cursor t the 25th line is cleared.
6 7				**************************************	ENTRY	none	······································
8 9 io				; 	EXIT	none	
10 11 12				; ;	USES	all	
13 14 OBAC 15 OBAE 16 OBB1 17 OBB3	E5 B4 F5 80	12 18 F0 EB	04	CLRS:	MÖV CJNE MÖV SJMP	A, YPOS A, #MAXLINE, CS1 E, A CLRL	; IF ypos = 25th line THEN ; setup for subroutine ; clear line and return
(8 19 OBB5 20 OBB7	7C 7A	<u>18</u> 00		CS1:	MOV MOV	WORK, #NUMLINES-1 INDX1, #O	; Get number of lines ; Index for -LORDER-
21 22 0889 23 0888 24 0880	8A 71 0A	Fó AO		CS2:	MOV ACALL INC	B, INDX1 CLRL INDX1	<pre>; REPEAT setup for call ; to clear a line ; bump ind×1</pre>
25 OBBE	DC	F9			DJNZ	WORK, CS2	; UNTIL work = 0
26 2 7 OB CO 28	81	36			AJMP	sch	; Set cursor home and return
31 32 33 34 35				7 7 		clear foreground clears entire display	except for half intensity spaces.
36 37 38				· · · · ;	ENTRY	none	
39				; .	EXIT	none	
40 41 42					USES	all	
43 44 0BC2 45 0BC5 46 0BC7	75 71 75	3F AC 3F	os oo	CLFO:	MOV ACALL MOV	PFIELD, #00001000B CLRS PFIELD, #0	; Make half intensity protected ; Clear screen ; Make nothing protected
46 OBC/ 47 OBCA	22	or 			RET	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	, have nothing riotected
				• • • • • • • • • • • • • • • • • • • •			

*.Z-29.COMPUTER.TER	051 CROSS ASSEMBL	ER (V1)-218 ERCION 1 02 *	24-FEB-83 15: **	:30:06 PAGE 114		
	70.40HC, 10.40HMHDE 1.50				ing of line	• • • • • • • • • • • • • • • • • • • •
3 4		; ;	*EBL*	erases from the ! • Cursor position	esinning of the current line to the	
5			ENTRY	none		
8 9						
10 11		, ,	USES	A, B, DPTR, TE	IP, PTR1, WORK2, INDX2	
13		EBL:	CLR	Α		
16 OBCF 12	12F0 0503		LCALL	B, YPOS CLA	; Calc address for (0,YPOS)	
17.08D2 E5 18.08D4 04 19.08D5 75	£0 Q0		MOV INC MOV	A, XPOS A B- #0	 Get number of characters to erase Clear attributes 	
20 OBD8 02 21	100C		LJMP	R2_FILL	; Clear and return	
22 23						
25 26					······	
27 28 29			*EEL* (Cursor	clears the entire Position does no	line where the cursor resides. t chanse.	•••••••••••••••••••••••••••••••••••••••
30 31			ENTRY	none		
32 33 34	•••••		EXIT	none		
36			USES	ABDPTRTE	P, PTR1, WORK2, INDX2	
37 38 OBDB 85 39 OBDE 80	12 F0 C0	EEL:	MOV SJMP	B, YPOS CLRL	; Get current position ; Clear line and return	
	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •			

()

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE 115 *** Z-29 COMPUTER TERMINAL FIRMWARE VERSION 1.03 *** *EOL* erases from the cursor to the end of the current line. Cursor position does not change. EXIT none MOV B, #0 ; Clear attributes 14 OBEO FO EOL: 50 EOL1: MÖV A, #NUMCHARS ; Calculate number to erase 16 OBE3 СЗ 95 CLR 17 OBE5 SUBB A, XPOS 18 OBE6 85 14 85 13 DPL, LINADL MOV 19 OBE8 .82 83 MOV DPH, LINADH 20 OBEB R2_FILL LUMP 21 OBEE 021996..... EBD - erase from beginning of display *EBD* clears the screen from 'HOME' position to the current cursor position. The exception is the 25th line. Cursor position does not chanse. EXIT.....none..... A, YPOS MOV EBD: 40 OBF1 E5 12 A. #MAXLINE, EB1 CUNE. ; IF 25th line THEN ; erase just this line .41 QBE3.... B4 SUMP EBL 42 OBF6 80 DЗ .43..... INDX1, #0 ; Start at top 44 OBF8 7A 00 EB1: MOV ; REPEAT EB2: A, YPOS MOV 12. .45 . QBEA. . ; IF index = current THEN A, INDX1A, EB3 CUNE 46 OBFC B5 02 02 ; erase up to cursor.... EBL 47 QBFF..... 80. CA SUMP 48 MOV -B, INDX1 ; ELSE set up and ; clear entire line ...49..0CQ1......\$A... CLRL 50 0003 71 AO ACALL bump index counter: INC INDX1 51 0C05 OA ; UNTIL forever EB2 SUMP F2 52 0006 80

					EED	erase to end of disp	lar
2 3				; •	*CCD*	-1 **	
7 4					tha an	d of the screen for	om the current cursor position to rsor position.
5					the en	a of the screen. Cu	ison rosition does not change.
5				;			
<u>7</u>					ENTRY	none	
3 9				•	CVIT		
"	• • • • • • • • • •				FXT1	none	
1				;	USES	A, B, DPTR, TEMP,	PTR1, INDX1, WORK2, INDX2
2							
3 **********	<u></u>	<u></u>		<u></u>			
0008	71	ΕO		EED:	ACALL		; Clear to end of line
OCOA OCOC	AA	12		EE1:	MOV	INDX1, YPOS	; Set loop counter
ocor	BA	18	00	CC1.	INC CUNE	INDX1	; REPEAT bump to next line
octo	<u>5</u>		00	EE1A:	JNC	INDX1, #MAXLINE, E	; IF end of disp THEN exit
0012	8A	FO		have been als 1 1 70	MOV	B, INDX1	; get line to clear
0014	71	Ao			ACALL	CLRL	clear line
0016	80	F4			SJMP	EE1	UNTIL forever
0018 3	····22			EER:	RÉT		
					• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
5							
,							
,				; ;	HEED -	Hazeltine erase to	end of display (with background)
7 3							end of diselay (with background)
,				5 5 5	*HEED*	clears the screen f	rom the current cursor position
,					*HEED*	clears the screen f	rom the current cursor position ith half intensity spaces. The
; ; ;				5 5 5 5 5	*HEED*	clears the screen f	rom the current cursor position ith half intensity spaces. The
,				7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	*HEED*	clears the screen f	rom the current cursor position ith half intensity spaces. The
3 3 3 3 3				5 5 5 5 7 7 7 7	*HEED*	clears the screen f	rom the current cursor position ith half intensity spaces. The
5 7 8 9 9 9				5 5 5 7 7 7 7 7	*HEED* to the cursor ENTRY	clears the screen f end of the screen w Position does not c	rom the current cursor position ith half intensity spaces. The
				7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	*HEED* to the cursor	clears the screen f end of the screen w Position does not c	rom the current cursor position ith half intensity spaces. The
				7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	*HEED* to the cursor ENTRY	clears the screen f end of the screen w position does not c none	rom the current cursor position half intensity spaces. The hanse.
; ; ; ; ; ;				7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	*HEED* to the cursor ENTRY	clears the screen f end of the screen w position does not c none	rom the current cursor position ith half intensity spaces. The
5				7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	*HEED* to the cursor ENTRY EXIT	clears the screen fend of the screen we position does not connone none A, B, DPTR, TEMP,	rom the current cursor position ith half intensity spaces. The hanse.
oc19	75	FO	08	;; ; ; ; ; ; ; ; ; ;	*HEED* to the cursor ENTRY EXIT USES	clears the screen fend of the screen we position does not control to the screen we have a screen with the screen for the screen we have a screen for the screen fend of the screen fend	rom the current cursor position ith half intensity spaces. The hange. PTR1, INDX1, WORK2, INDX2 ; Get half intensity
oc19	75 71	Fo E3	08	;; ; ; ; ; ; ; ; ; ; ; ;	*HEED* to the cursor ENTRY EXIT	clears the screen fend of the screen we position does not connone none A, B, DPTR, TEMP,	rom the current cursor position ith half intensity spaces. The hanse.
oc19	75 71	F0 E3	08	;; ; ; ; ; ; ; ; ; ; ; ;	*HEED* to the cursor ENTRY EXIT USES MOV ACALL	clears the screen fend of the screen we position does not contain the screen we have a second contained by the screen we have a second contained by the screen was a second contained by the screen we have a second contained by the screen we have a second contained by the screen was a second contained by the screen we have a second contained by the screen was a second contained by	rom the current cursor position ith half intensity spaces. The hange. PTR1, INDX1, WORK2, INDX2 ; Get half intensity ; Clear to end of line
oc19	71	Ę3	08	;; ; ; ; ; ; ; ; ; ; ; HEED:	*HEED* to the cursor ENTRY EXIT USES MOV ACALL	clears the screen fend of the screen we position does not continue. none A, B, DPTR, TEMP, B, #00001000B EOL1 INDX1, YPOS	rom the current cursor position ith half intensity spaces. The hanse. PTR1, INDX1, WORK2, INDX2 ; Get half intensity ; Clear to end of line ; Set loop counter
0019 0019 0010 0016	71 AA	12 13	08		*HEED* to the cursor ENTRY EXIT USES MOV ACALL	clears the screen fend of the screen we position does not connote. none A, B, DPTR, TEMP, B, #00001000B EOL1 INDX1, YPOS INDX1	rom the current cursor position ith half intensity spaces. The hange. PTR1, INDX1, WORK2, INDX2 ### Get half intensity ### Clear to end of line ### Set loop counter ### ### ### ### ### ### ### #### ####
0019 0019 0019 0016 0020	71 AA OA BA 50	12 13			*HEED* to the cursor ENTRY EXIT USES MOV ACALL MOV INC	clears the screen fend of the screen we position does not continue. none A, B, DPTR, TEMP, B, #00001000B EOL1 INDX1, YPOS	rom the current cursor position ith half intensity spaces. The hange. PTR1, INDX1, WORK2, INDX2
0C19 0C19 0C19 0C10 0C20 0C21	71 OA OA BA 50 E4	É3 12 18 F2		HEE1:	*HEED* to the cursor ENTRY EXIT USES MOV ACALL MOV INC CUNE JNC CLR	clears the screen fend of the screen was position does not consider the screen was position does not consider the screen was positioned. A, B, DPTR, TEMP, B, #00001000B EOLI INDX1, YPOS INDX1 INDX1, #MAXLINE, HEER A	rom the current cursor position ith half intensity spaces. The hange. PTR1, INDX1, WORK2, INDX2 ### Get half intensity ### Clear to end of line ### Set loop counter ### ### ### ### ### ### ### #### ####
0C19 0C19 0C1C 0C1C 0C20 0C21 0C24 0C24	71 OA OA BA 50 E4 8A	E3 12 18 F2 F0		HEE1:	*HEED* to the cursor ENTRY EXIT USES MOV ACALL MOV INC CONE JNC CLR MOV	clears the screen fend of the screen we position does not continue. none A, B, DPTR, TEMP, B, #00001000B EOLI INDX1, YPOS INDX1 INDX1, #MAXLINE, H EER A B, INDX1	rom the current cursor position ith half intensity spaces. The hange. PTR1, INDX1, WDRK2, INDX2 ; Get half intensity ; Clear to end of line ; Set loop counter ; REPEAT bump to next line EEIA ; IF end of disp THEN exit ; set line to clear
0C19 0C19 0C1C 0C1C 0C20 0C21 0C24 0C24 0C27	71 OA OA BA 50 E4 8A	E3 12 18 F2 F0 0503		HEE1:	*HEED* to the cursor ENTRY EXIT USES MOV ACALL MOV INC CUNE JNC CLR MOV LCALL	clears the screen fend of the screen we position does not connote. none A, B, DPTR, TEMP, B, #00001000B EOL1 INDX1, YPOS INDX1 INDX1, #MAXLINE, HEER A B, INDX1 CLA	rom the current cursor position ith half intensity spaces. The hange. PTR1, INDX1, WORK2, INDX2 ; Get half intensity ; Clear to end of line ; Set loop counter ; REPEAT bump to next line EEIA ; IF end of disp THEN exit ; get line to clear ; calculate line address
0019 0019 0010 0020 0024 0024 0024 0027	71 AA OA BA 50 E4 8A 12 74	12 18 18 F2 F0 0503 50		HEE1:	*HEED* to the cursor ENTRY EXIT USES MOV ACALL MOV INC CUNE JNC CLR MOV LCALL MOV	clears the screen fend of the screen we position does not continue. none A, B, DPTR, TEMP, B, #00001000B EOL1 INDX1, YPOS INDX1 INDX1, #MAXLINE, HEER A B, INDX1 CLA A, #NUMCHARS	rom the current cursor position ith half intensity spaces. The hange. PTR1, INDX1, WORK2, INDX2 ; Get half intensity ; Clear to end of line ; Set loop counter ; REPEAT bump to next line EEIA ; IF end of disp THEN exit ; get line to clear ; calculate line address ; complete line
0019 0010 0010 0010 0010 0020 0024 0024 0027 0029 0029	71 AA OA BA 50 E4 8A 12 74	12 18 F2 F0 0503 50 F0		HEE1:	*HEED* to the cursor ENTRY EXIT USES MOV ACALL MOV INC CUNE JNC CLR MOV LCALL MOV MOV MOV	clears the screen fend of the screen we position does not continuous none none A, B, DPTR, TEMP, B, #00001000B E0L1 INDX1, YPOS INDX1 INDX1, #MAXLINE, HEER A, B, INDX1 CLA A, #NUMCHARS B, #00001000B	rom the current cursor position ith half intensity spaces. The hange. PTR1, INDX1, WORK2, INDX2 ; Get half intensity ; Clear to end of line ; Set loop counter ; REPEAT bump to next line EEIA ; IF end of disp THEN exit ; get line to clear ; calculate line ; get half intensity
0019 0019 0019 0010 0020 0021 0024 0024 0027 0029 0020 0026	71 AA OA BA 50 E4 8A 12 74 75	E3 12 18 F2 F0 0503 50 F0 100C		HEE1:	*HEED* to the cursor ENTRY EXIT USES MOV ACALL MOV INC CUNE JNC CLR MOV LCALL MOV LCALL	clears the screen fend of the screen we position does not continue. none A, B, DPTR, TEMP, B, #00001000B EOL1 INDX1, YPOS INDX1 INDX1, #MAXLINE, HEER A B, INDX1 CLA A, #NUMCHARS B, #00001000B R2_FILL	rom the current cursor position ith half intensity spaces. The hange. PTR1, INDX1, WORK2, INDX2 ; Get half intensity ; Clear to end of line ; Set loop counter ; REPEAT bump to next line EEIA ; IF end of disp THEN exit ; set line to clear ; calculate line address ; complete line ; set half intensity ; fill with spaces
0019 0010 0010 0010 0010 0020 0024 0024 0027 0029 0029	71 AA OA BA 50 E4 8A 12 74	12 18 F2 F0 0503 50 F0		HEE1:	*HEED* to the cursor ENTRY EXIT USES MOV ACALL MOV INC CUNE JNC CLR MOV LCALL MOV MOV MOV	clears the screen fend of the screen we position does not continuous none none A, B, DPTR, TEMP, B, #00001000B E0L1 INDX1, YPOS INDX1 INDX1, #MAXLINE, HEER A, B, INDX1 CLA A, #NUMCHARS B, #00001000B	rom the current cursor position ith half intensity spaces. The hange. PTR1, INDX1, WORK2, INDX2 ; Get half intensity ; Clear to end of line ; Set loop counter ; REPEAT bump to next line EEIA ; IF end of disp THEN exit ; get line to clear ; calculate line ; get half intensity

C

1				scH -	set cursor home	
2			<u>.</u>	*SCH*	places the cursor	at line zero, column zero.
4 5			; 			······································
6 7			; ;	ENTRY	none 	
9			; ;	EXIT	none	
(ö L1			;	USES	A, DPTR, PTRI	
12 13 0036	E4 F5		sch:	CLR	A	
14 0037 15 0039	F5	11		MOV MOV	XPOS, A YPOS, A	; Zero X-position ; Zero Y-position
[4003B]	. F5 . 21	12 +4		AMEA	XBLINAD	; Updata address and return
l 7 [8						
19 20					· · · · · · · · · · · · · · · · · · · ·	
21 22				PBS -	perform back space	
22 23 24			,	CLFT -	cursor left	
2 4 25 26			; ;	*PBS*	- *CLFT* steps the	cursor one position to the left, but
26 27			# · · · · · · · · · · · · · · · · · · ·	does n	ot wrap around to zero.	the previous line after reaching
27 28 29			# #	ENTRY	none	
29 30 31			,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	EXIT	none	
32 33			*	USES	A, DPTR, PTR1	
34						
35 36	.003D		PBS CLFT	EQU EQU	\$ \$	
37 38 OC3D	.0C3D	11		MOV	A, XPOS	; Get X-position
39 OC3F 40 OC41	. 70 22	O1		JNZ RET	PBS1	; IF -XPOS- is zero THEN ; return
41	15	11	PBS1:	DEC	XPOS	; ELSE decrement -XPOS-
42 OC42	21	F4		AJMP	XBLINAD	; build address and return

)))))))

·\$·····					ACLET	- ANSI cursor left	
.ş̃					*ACLFT	* moves the cursor t	owards the beginning of a line
.5	· · · · · · · · · · · · · · · · · · ·			,	the sp	ecified Pn number of	times.
6 .7				;	ENTRY	none	
8 .9 .0				;	EXIT	none	
1				,	USES	A, DPTR, INDX1, PI	R1
(2 (3							
4 0C46 5 0C48	31 70	Di		ACLFT:	ACALL	GNP	; Get parameter
.6 004A	64	01		• • • • • • • • • • • • • • • • • • • •	JNZ I NC	ACLO.	; IF no parameters or O THEN ; bump to 1
.7 [8] [664 8] [FA			ACLO:	MOV	INDX1, A	; Set up index
.9 0040 20 004611	91 DA	3D FC		ACL:	ACALL	CLFT	; REPEAT move cursor left one
21 0050	22	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,			DJNZ RET	INDX1, ACL	; UNTIL count = 0
2 3 4							
:5 :6				; ;	HCLFT	- Hazeltine cursor 1	eft
?7. :8				•••••	*HCLFT	* steps the cursor o	ne position to the left and wraps
:9 :0					around	to the previous lin	e after reachins column zero.
1 2					ENTRY	none	
:3 :4:					EXIT		
5 5				, , , , , , , , , , , , , , , , , , ,		none	
7				,	USES	"A, DPTR, PTR1, TEM	P
9 OC51	E5	11		HCLFT:	MOV	A, XPOS	; Get X-position
ბ∵ბႠ\$3∵ 1 0C55	70 AD	E3			JNZ	CLFT	; IF not zero THÉN use *CLFT*
2 0057 ··	gi	12	• • • • • • • • • • • • •		MOV ACALL	TEMP, YPOS	; Save Y-poșition ; Move cursor up
3 0059	E5	12			MOV	A, YPOS	, move cursor ar
4 005B	B5	05	61		CÛNE	"A, TEMPA, HOLFT1"	; IF yeos did not change THEN
5 OC5E 6	22	• • • • • • • • • • • • • • • • • • • •			RET		; do nothins
7 OC5F 81006211	75 21	f1	4F	HCLFT1:	MOV MCA	XPOS, #MAXCHAR XBLINAD	; ELSE move to end of line ; update and return
	 					VPCIMAN	, update and return
					• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	••••••
							·····

.1					CRT -	ursor right	
2 3				; ;	*CRT* i	noves the cursor right	one column. If the cursor is in
4 5				;	the la	st column then the curs	sor will not be advanced.
ر اد				;	ENTRY	none	
8 9 10				; ;	EXIT		
10 11				; ;		A, DPTR	
12 13							
14 0064 15 0066	E5	1 1 4F	01	CRT:	MOV	A, XPOS A, #MAXCHAR, CRT1	; Get X-position ; IF end of line THEN
16 0069	<u>B4</u>		Y. .		CJNE RET		; return
17 18 006A	ö5	11		CRT1:	INC	XPOS	; ELSE bump position
19 0060 20 21	21	F.4			AJMP	XBLINAD	
.41 22 .23							
24 25		.,	• • • • • • • • • • • • • • • • • • • •	, , , , , , , , , , , , , , , , , , ,	ACRT -	ANSI cursor right	
26 27				; ;	*ACRT* the se	moves the cursor towar ecified Po number of ti	ds the end of the line mes.
28	,			,			
29 30 31				.	ENTRY	none	
32			• • • • • • • • • • • • •	· · · · · · · · · · · · · · · · · · ·	EXIT	none	
33 34 35	,]	USES	A, DPTR, INDX1	
36					ACALI	CND	. C-bb
37.004E 38.0070	31 70	D1 01		ACRT:	AÇALL JNZ	ACR1	; Get parameter ; IF no parameter or O THEN
32.9072 40	94				INC	A	; bume to 1
41. QCZ3	F.A			ACR1:	MOV	INDX1. A	; Set up index
42 0074 43 0076	91 DA	64 FC		ACR:	ACALL DJNZ	CRT INDX1, ACR	; REPEAT cursor right one ; UNTIL count = O
3.0074 4.0078	DA 22	F.C			DUNZ RET	INDX1, ACR	; UNTIL count = 0
					• • • • • • • • • • • • • • • • • • • •		

)

2 #MCRT#.moxes.the_cursor.risht.ane_columnIf.the_cursor.is. 4 in the last column then the cursor moves to the besinning of 5 the_bost_line. 7 the_bost_line. 8 ENTRY none 9 the last result in the last column then the cursor moves to the besinning of 10 EXIT none 11 EXIT none 11 USES A. DPTR. PTRI 12 USES A. DPTR. PTRI 13 the last result in the last	3		. 		5	HÇRT	Hazeltine.cumson.misht	i	
1					;				
S	4				;	in the	last column then the o	ursor moves to the beginning of	
S	6				•				
9	7		• • • • • • • • • • • • • • • • • • • •			ENTRY			
10			• • • • • • • • • • • • • • • • • • • •				•••••		
12 ; USES A. DFTR, PTR1 13		· · · · · · · · · · · · · · · · · · ·			; ;				
15. 0C79 E5. 11 HCRT! MOV A. XPOS I. Get X-Position. 16 0C7B B4 4F E6 CUNE A. WHACHAR. CRT I IF not eol THEN use *CRT* 17. 0C7E AB. 12 MOV TEMP. YPOS I. Save Y-Position. 18 0C80 91 A8 ACALL CDN I MOVE cursor down 19. 0C82 E5. 12 MOV A. YEOS. 20 0C84 B5 05 01 CUNE A. TEMPA, HCRTI I F xpos did not chanse THEN 21. 0C87 22 RET I do.nothins 22 RET I DO. I MOVE CONTROL TO THE TEMPA CONTROL THE TEMPA CONTROL TO THE TEMPA CONTROL	12				5	USES	A, DPTR, PTR1		
16 OC7B B4 4F E6 CUNE A HYAKCHAR, CRT : IF not eol THEN use SCRT* 17, OC7E AD 12 MOV TEMP, YPOS : Save Y-resition 18 OC80 91 A8 ACALL CDN ; Move cursor down 19, OC82 E5, 12 MOV. A, YPOS ; IF xeos did not chanse THEN 21, OC87, 22 RT ; do, nothing 20 OC84 B5 O5 O1 CUNE A TEMPA, HCRT1 : IF xeos did not chanse THEN ET1 ; do, nothing 21, OC87, 22 RT ; do, nothing 22 CS, T5, 11 QO HCRT1; MOV XPOS, #0 IS ELSE move to beginning 24 OC8B 21 F4 AUMP XBLINAD : update and return	14								
.17.0C7E AD 12 MOV. TEMP. YPOS ; Save Y-resition. 18.0C80 91 A8 ACALL CDN ; Move cursor down. 19.0C82 E5 12 MOV. A. YPOS ; Move cursor down. 20.0C94 B5 05 01 CUNE A. TEMPA, HCRT1 ; IF xpos did not change THEN ; 1.0C87 22 RET. ; 30.0C10.0E2.			11 4F	E6	HÇRT	MQY			• • • • • • • • • • • • • • • • • • • •
19.0C82		ζĘAD	12			MQV	TEMP: YPOS		
20 0084 BS 05 01 CJNE A, TEMPA, HCRT1 ; IF xpos did not chanse THEN 21,0087, 22 REI i.do. nothing 22 22						ACALL MOV		; Move cursor down	
22 OC88 75 11 00 HCRT1: MOV XPOS, #0 : ELSE mave to beginning 24 OC8B 21 F4 AJMP XBLINAD ; urdate and return	20 008	34 BS	05	01		CJNE	A, TEMPA, HCRT1	; IF xpos did not change THEN	
24 OCBB 21 F4 AJMP XBLINAD ; update and return	22								
	23.009	3875		QQ	HCRT1:	MQV	.XPOS, #0		
		••••••	••••••						
······································		• • • • • • • • • • • • • • • • • • • •		•••••					
······································									
······································		······································							
······································									

(

C

((

1				; ;	CUP -	ursor up	
3				5	*CHP*	novee the cureor up or	e line on the display. The
3 4	• • • • • • • • • • • • • • • • • • • •			 •	cursor	may move out of but r	ot into fixed regions and
5				<u>.</u>		Past line zero.	
5 7				,			
3				5	ENTRY	none	
9 		• • • • • • • • • • • • • • • • • • • •		·•••••••••••••••••••••••••••••••••••••	EXIT	none	
1 2				<u>.</u>	USES	A, DPTR, PTR1	
2 3				•	USES	H, DEIK, FIKI	
4				CUD.	MOV	A VDOC	* Cat V-magitian
5 0C8D 6 0C8F	<u>E5</u>	12		CUP:	JŽ	A, YPOS CUP1	; Get Y-position ; IF top of screen THEN return
7 0091		17	01			A, TSCROLL, CUP1A	; IF top of scroll region THEN
7. 0021 3 0094	B5 22		¥#	CUP1:	RET		f return
9 0 0095	 В4	18	01	CUP1A:	CUNE	A, #MAXLINE, CUP1B	; IF 25th line THEN
) 0095 1.0098	22	18	01	CUPIA:	RET	H, #PHALINE, COPID	; return
2							
3.QC22. 4 OC2B	15 21	12 F4		CUP1B:	DEC	YPOS XBLINAD	; ELSE dec Position; update and return
ā							
5 7							
3					ACUD	ANCT SUBSESSION	
₹ o				.7. ?	AGUP	ANSI cursor up	
ī 1				5			and the top of the display
2 3				;	the sp	ecified Pn number of t	imes.
4			•••••	3			
5 5					ENTRY	none	
- 7					EX.T	none	
3				;	uece	A. DPTR. INDX1. PTRI	
? o				.7			
1	31	D1		ACUP:	ACALL	GNP	; Get parameter
2 OC9D 3 QC9E	31 ZQ	01		HUUF.	JNZ	ACU1	
4 OCA1	04	· · · · · · · · · · · · · · · · · · ·			INC	A	; bump to 1
5 5 OCA2	FA	• • • • • • • • • • • • • • • • • • • •		ACU1:	MOV	INDX1, A	; Set up index
5 OCA2 7.0ÇA3	9.1	8D		AÇU:	AÇALL.	CUP	; REPEAT cursor up one line
B OCA5	DA	FC			DUNZ	INDX1, ACU	; UNTIL count = 0
?QQA7	22				RET		
							
		• • • • • • • • • • • • • • • • • • • •					
							

Z 3 4 5 5					cursor down	
4 5 6			; ;	*CDN*	moves the cursor do	wn one line on the display. The
ā 6 7			,	cursor	may move out of bu	it not into fixed regions and
 7	• • • • • • • • •			nexer.	rast.line.24	
'.						
3 3			;	ENTRY	none	
ó	• • • • • • • • •		# # # # # # # # # # # # # # # # # # #	EXIT	none	
<u>1</u> 2				ÜŚEŚ	A, DPTR, PTR1	
3 4						
4 5 OCA8	E5	12	CDN:	MOV	A, YPOS	; Get Y-position
S OCAA	E5 	12 18	01	CÜNE	A, #MAXLINE, CONI	
7 OCAD	22			RET		
OCAE	В4	17	01 CDN1:	CUNE	A, #MAXLINE-1, CI	N2 ; IF 24th line THEN return
O OCB1	22			RET		
1 2 OCB2	04		CDN2:	INC	A	; Bump value
3 OCB3 4 OCB6	B5 22	18	01	CJNE RET	A, BFIX, CDN3	; IF scroll region THEN return
5 5 5 OCB7						,
5 OCB7	05 21	12	CDN3:	INC	YPOS	; ELSE bump position
7.0CB9		F4	•••••	AJMP	XBLINAD	; Update and return
1 2 3 4			; ; ;		ANSI cursor down	
5 			; ;		moves the cursor t ecified Pn number o	oward the bottom of the display of lines.
7 3	• • • • • • • • • • • • • • • • • • • •		· · · · · · · · · · · · · · · · · · ·	······ENTRY	none	
? `					_	
l			, 5	EXIT	none	
2····· 3	• • • • • • • • •		,	USES	A, DPTR, INDX1, F	TRI
¥·····	. 	• • • • • • • • • • • • • • • • • • • •				
OCBB	31 7 0	D1	ACDN:	ACALL JNZ	GNP ACD1	; Get parameter
OCBF	04	01		INC	ACD1 A	; IF no parameters or 0 THEN ; bump to 1
3 0000			0004+			
0.0001 0.0001	FA 91	A8	ACD1:	MOVACALL	INDX1, A	; Set up index ; REPEAT cursor down one line
0003	DA	FC		DJNZ	INDX1, ACD	; UNTIL count = 0
2 0005	22			RET		
	• • • • • • • • •				• • • • • • • • • • • • • • • • • • • •	•••••••••••••••••••••••••••••••••••••••
		• • • • • • • • • • • • • • • • • • • •				

! 2	; ;	APBT -	ANSI perform back	tab
	• • • • • • • • • • • • • • • • • • • •			
		APBT	moves the cursor b	ack the number of tab stops indicated
5 6 -		ENTRY	none	
7 8		EXIT	none	
9 10 11		USES	all	
11 12 13 0006 31 D1	APBT:	ACALL	GNP	; Get parameter
13 OCC6 31 D1 14 OCC8 70 01 15 OCCA 04		ACALL JNZ INC	APBT1 A	; Get parameter ; IF no parameter or O THEN ; bump to 1
16	APBT1:		ACC BTAB	; REPEAT ; perform back tab
18 OCCD 12 0602		PUSH LCALL POP	BTAB ACC	
19 0CD0 D0 E0 20 0CD2 D5 E0 F0 21 0CD5 22		POP DJNZ RET	ACC, APBT1	; UNTIL count = 0
BA . NYBEY				

<u>1</u>					SVA	cursor.addressina.routin	ę
4 3					*CUA*	does direct cursor addre	ssins for a few emulation modes.
4 5				;			· · · · · · · · · · · · · · · · · · ·
6					ENTRY	none	·····
/ 8					EXIT	none	
9 					USES	***************************************	
1				,	USES	all	
2 3 OCD6	11	2A		CUA:	ACALL	SETDISP	A Code discount of the last
4			• • • • • • • • • • • • • • • • • • • •	COM*		SEIDISF	; Set dispatch and return
5 OCDB 6 OCDB	75 75	<u>16</u>	<u>04</u>		MOV MOV	DSADRL, #LOW (CUC)	; Next dispatch address
7 OCDE	30	43	11		JNB	DSADRH, #HIGH (CUC) HAZ, CUBO	TE Hamalada a Then
8 OCE1	B4	60	····· ôō····	• • • • • • • • • • • • • • • • • • • •	CJNE	A, #96, CUA1	; IF Hazeltine THEN
9 OCE4	40	02		CUA1:	JC	CUA2	; IF value >= 96 THEN
0 OCE6	94	60			SUBB	A, #96	; subtract offset
1 OCES 2 OCEB	B4 40	50 02	00	CUA2:	CJNE	A, #NUMCHARS, CUA2A	
3 OCED	74	02 4F		CUA2A:	JC MOV	CUA3	; IF value >= numchars THEN
4 OCEF	<u>:</u>	·····: ? [2F		CUA3:	SJMP	A, #MAXCHAR CUR2	, move to end of line
5							; update and return
5 OCF1 7	22			CUR:	RET		
0CF2	C3			CUBO:	CLR	·····c	
9 OCF3	94	29			SUBB	. A., #1. 1	; ELSE subtract offset
0 0CF5 1 0CF7	40 B4	FA	0.4		JC	CUR	; check for lower bounds
OCFA	54	18 15	04 F4		CJNE JNB	A, #MAXLINE, CUB1 L25EN, CUR	; check for upper bounds
3 OCFD	D3	10	17		SETB	C C	; IF 25th disabled THEN ret ; ELSE fall through
4 OCFE	5ò	Fi		CUB1:	JNC	CUR	; IF upper bounds ok THEN
5 ODOO	F5	12		CUR1:	MOV	YPOS, A	; replace Y-position
5 0002 '' 7	21	F4			AJMP	XBLINAD	; build line address
3 ODO4	11	2F		·····cuc:	ACALL	SETNORM	; Next dispatch address
OD06	30	43 1F	OB		JNB	HAZ, CUCO	; IF Hazeltine THEN
0009	54				ANL	A, #00011111B	; do mod function
LODOB 2 ODOB	<u>84</u>	<u>1</u> 8	00		CUNE	A, #MAXLINE, CUC1	
3 OD0E	40 74	02 17		cuci:	JC MOD	CUCIA	; IF value >= 24 THEN
. OD10	ģõ	····· Ėć·····		CUCTA:	MOV SJMP	A, #MAXLINE-1 CUR1	move to last line
;					30111		; update and return
OD14				cuco:	CLR	Ċ	
' OD15	94	20			SUBB	A, #1 1	; Subtract offset
0017 0019	40 B4	05 50	00		JC	CUC2	; Subtract offset ; Check for lower bounds
OD16	<u>84</u>	50 02	99	CUCOA:	CJNE JC	A, #NUMCHARS, CUCOA	***************************************
OD1E	74	4F		CUC2:	MOV	CUR2 A, #MAXCHAR	Check for upper bounds
OD20	<u>ŕ</u> š			CUR2:	MOV	XPOS, A	; IF bad range THEN move to end ; Replace X-position
OD22	21	F4		· · · · ·	AJMP	XBLINAD	, Replace X-Position .; Build line address and return
							······································

(((

	Second S	Second State Sec	1	• • • • • • • • • • • • •			;;	DSR -	define scrolling regio	on
The region must be at least two lines long and can not include the ZSth line.	The region must be at least two lines long and can not include to the 25th line. The region must be at least two lines long and can not include to the 25th line. The region must be at least two lines long and can not include the the 25th line. The region must be at least two lines long and can not include the the 25th line. The region must be at least two lines long and can not include the the 25th line. The 25th li	The region must be at least two lines long and can not include the 25th line. The region must be at least two lines long and can not include the 25th line. The region must be at least two lines long and can not include the 25th line. The region must be at least two lines long and can not include the 25th line. The region must be at least two lines long and can not include the the 25th line. The 25th	2				;	xnepx	defines the ecrolling	regions top and hottom line
The 25th line State Stat	The 25th line Section Section	S					💃	The re	gion must be at least	two lines long and can not include
ENTRY	ENTRY None	6	5				•			
ENTRY	ENTRY	S	_		• • • • • • • • • • • • • • • • • • • •		***************************************			
Second	Second	Second								
	STIT None STIT STIT None STIT STIT None STIT	10					;	ENTRY	none	
1	1	12	. X		• • • • • • • • • • • • • • • • • • • •			FXIT	none	
3	3	13	11				;			
		14	12				· · · · · · · · · · · · · · · · · · ·	USES	A, DPTR	
15 0D24 31 D1		15 0024 31 01 05R: ACALL GNP								
DEC A	DEC A	17 0D28	14 15 ADO/		D.1		nep.	۸۵۸۱۱	GNP	: Got first parameter
DEC A	DEC A	17 0028 14 DEC A	15 0024	51	·····		həv.			: IF parameter and not O THEN
18	18	18 19 0D29			0.			DEC		; normalize to 024
MOV	MOV TSCROLL; A	21 ODZE F5 17 MOV TSCROLL, A ; change value 22			• • • • • • • • • • • • • • • • • • • •					
MOV	MOV TSCROLL; A	21 ODZE F5 17 MOV TSCROLL, A ; change value 22	19 OD29	, В4	17	99	DSRO:		A. #MAXLINE-1, DSROA	f
Dec	Dec	22 23 0D30	20 OD20		02		DSROA:	JNC	DSR1	
DSR DSR	23 0D30 31 D1 DSR1: ACALL GNP ; Get next parameter 24 0D32 60 01 JZ DSR1A ; IF parameter and not 0 THEN 25 0D34 14 DEC A ; normalize to 024 27 0D35 B5 17 01 DSR1A: CJNE A, TSCROLL, DSR2 28 0D38 D3 SETB C ; IF bottom = top OR 29 29 20 DSR2: JC DSRERR ; IF bottom < top OR 31 0D3B B4 18 00 CJNE A, #MAXLINE, DSR2A 32 0D3E 40 02 DSR2A: JC DSROK ; IF bottom >= 25 THEN error 33 34 0D40 74 17 DSRERR: MOV A, #MAXLINE-1 ; Default error value 35 0D42 04 DSROK: INC A ; Bump for top of bottom fixed 36 0D43 F5 18 MOV BFIX, A ; Save value	23 0D30	21OD2E	F5	17			MOV	TSCROLL, A	i. Cyduae Agine
JZ	24 0D32 60 01	24 0D32 60		. 21	D.		nep1•	ACAL I	GND	: Get peyt parameter
DEC A ; normalize to 024 26 27 OD35	DEC A ; normalize to 0,.24 26 27 OD35	25 OD34 14 DEC A ; normalize to 024 26 27 OD35 B5 17 O1 DSR1A: CJNE A, TSCROLL, DSR2 28 OD38 D3 SETB C ; IF bottom = top OR 29 30 OD39 40 O5 DSR2: JC DSRERR ; IF bottom < top OR 31 OD3B B4 18 00 CJNE A, #MAXLINE, DSR2A 32 OD3E 40 O2 DSR2A: JC DSROK ; IF bottom >= 25 THEN error 33 34 OD40 74 17 DSRERR: MOV A, #MAXLINE-1 ; Default error value 35 OD42 O4 DSROK: INC A ; Bump for top of bottom fixed 36 OD43 F5 18 MOV BFIX, A ; Save value	24 OD30	'≳¥			N5U1.		DSR1A	: IF parameter and not O THEN
26 27 OD35 B5 17 O1 DSR1A: CJNE A, TSCROLL, DSR2 28 OD38 D3 SETB C ; IF bottom = top OR 29 30 OD39 40 O5 DSR2: JC DSRERR ; IF bottom < top OR 31 OD3B B4 18 OO CJNE A, #MAXLINE, DSR2A 32 OD3E 40 O2 DSR2A: JC DSROK ; IF bottom >= 25 THEN error 33 34 OD40 74 17 DSRERR: MOV A, #MAXLINE-1 ; Default error value 35 OD42 O4 DSROK: INC A ; Bump for top of bottom fixed 36 OD43 F5 18 MOV BFIX, A ; Save value	26 27 OD35	26 27 0D35	25 OD34	14	٠.			DEC		normalize to O24
28 OD38 D3 SETB C ; IF bottom = top OR 29 29 30 OD39	28 OD38 D3 SETB C ; IF bottom = top UR 29 29 29 30 OD39	28 OB38 D3 SETB C ; IF bottom = top OR 29 29 30 OB39	26							
29 30 0D39	29 30 0D39	29 30 0D39	27. OD35	jB5	17	01	D\$R1A:	CUNE	A, TSCROLL, DSR2	
30 OD39	30 OD39	30 0D39		; D3				SETB	С	; IF bottom = top UR
31 OD3B B4 18 00 CJNE A, #MAXLINE, DSR2A 32 OD3E 40 02 DSR2A: JC DSROK ; IF bottom >= 25 THEN error 33 OD40 74 17 DSRERR: MOV A, #MAXLINE-1 ; Default error value 35 OD42 04 DSROK: INC A ; Bump for top of bottom fixed 36 OD43 F5 18 MOV BFIX, A ; Save value	31 OD3B	31 OD3B B4 18 Q0 CJNE A, #MAXLINE, D\$R2A 32 OD3E 40 O2 D\$R2A: JC D\$ROK ; IF bottom >= 25 THEN error 33 OD40 74 17 D\$RERR: MOV A, #MAXLINE-1 ; Default error value 35 OD42 Q4 D\$ROK: INC A ; Bump for top of bottom fixed 36 OD43 F5 18 MOV BFIX, A ; Save value	29 20. opas				nepo:		nseree	: IF bottom < top OR
32 ODSE 40 O2 DSR2A: JC DSROK ; IF bottom >= 25 THEN error 33 34 OD4O 74 17 DSRERR: MOV A, #MAXLINE-1 ; Default error value 35 OD42 O4 DSROK: INC A ; Bump for top of bottom fixed 36 OD43 F5 18 MOV BFIX, A ; Save value	32 OD3E 40 O2 DSR2A: JC DSROK ; IF bottom >= 25 THEN error 33 34 OD4O 74 17 DSRERR: MOV A, #MAXLINE-1 ; Default error value 35 OD42 04 DSROK: INC A ; Bume for top of bottom fixed 36 OD43 F5 18 MOV BFIX, A ; Save value	32 OD3E 40 O2 DSR2A: JC DSROK ; IF bottom >= 25 THEN error 33 34 OD4O 74 17 DSRERR: MOV A, #MAXLINE-1 ; Default error value 35 OD42 O4 DSROK: INC A ; Bume for top of bottom fixed 36 OD43 F5 18 MOV BFIX, A ; Save value	30 OD37 31 OD3F	3 B4		00		CJNE		
34 OD4O	34 OD4O	34 OD4O	32 OD3E	40			DSR2A:	JC		; IF bottom >= 25 THEN error
35 OD42	35 OD42	35 OD42							<u>.</u>	
35 0D42 04 DSRON: AND H 1 BUND TO TO TO THE TREE OF SECTION 1 AS 0D42 04 1 Save value 36 0D43 F5 18 MOV BFIX, A 1 Save value 37 0D45 81 36 AJMP SCH 1 Set cursor home and return	35. 0D42	35, 0042 P5 18 MOV BFIX, A ; Save value 36 0043 P5 18 AJMP SCH ; Set cursor home and return		74	17					
AJMP SCH : Set cursor home and return	AJMP SCH i Set cursor home and return	37 OD45 81 36 AJMP SCH 1 Set cursor home and return	30.QU42							
				, FJ 5 81	36				SCH	
			AC. YETE							
					• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •			

)))))))

)

				, 5, 5	PIL - r	erform insert line	
2 3							t the cursor position after
4				·····;		the remaining lines do	own one line. Everything is
5							the cursor is in, top fixed,
6				;	bottom	fixed, 25th line, or	scrollina reaion. No reaion
·····	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • •		outside	of the current resions the	n is effected. The cursor is
9				, ;	moved (o the peathnites of the	current line.
10				;		• • • • • • • • • • • • • • • • • • • •	
<u>11</u> 12					ENTRY	none	***************************************
13				; :	EXIT	none	
14			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		none	
					USES	all	
15 17							
17 18 OD47	ćò	iż		PIL:	PÜŚH	YPOS	• Cava values
19 OD49	co			I Alm.	PUSH	TSCROLL	; Save values
20 OD4B	co	17 18	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	PUSH	BFIX	•••••••
21 OD4D 22 OD4F	<u>E5</u>	12 17			MOV	A, YPOS	; Get position
22 OD4F 23 OD52	B5 40	17 21	00	DT. 0.	CUNE	A, TSCROLL, PILA	; IF upper fixed region THEN
24 OD54				PILA:	JC DEC	PILRT1 BFIX	; just return
25 OD56 26 OD59	B5	18	67 F0		CUNE	A, BFIX, PIL1	; IF last line scrollina region
	85	12	FO		MOV	B, YPOS	
27 OD5C 28 OD5E	<u>71</u>	<u>96</u>		PILO:	ACALL	CLRLINE PILRET	; just clear the line
29	00				SUMP	PILKEI	
30.0000	40		• • • • • • • • • • • • • • • • • • • •	PILI:	JC	PIL2	; IF lower fixed THEN
31							
32 OD62 33 OD65	84 75	18 F0	10		CUNE MOV	A, #MAXLINE, PILRT1	; IF not 25th line THEN return
34.0088.	ģŏ	·····¦rž·····	18	• • • • • • • • • • • • • • • • • • • •	SJMP	B, #MAXLINE PILO	; ELSE clear 25th line ; and return
35						. 120	, and return
36 006A	05	18		PIL2:	1NC	BFIX	•••••••••••••••••••••••••••••••••••••••
37 OD6C 38 OD6F	85 i2	12 0518	17		MOV	TSCROLL, YPOS	; Insert line uses
રવ	12	0518			CALL	PRLF	; Reverse line feed routine
	····75	ii		PILRET:	MOV	XPOS, #0	; Move to beginning of line
41 OD75	Do	18		PILRT1:	POP	BFIX	Restore values
42 0077 43 0079	DO	<u>ī</u> j			POP	TSCROLL	***************************************
43 OD79	<u>D</u> O 21	<u>12</u>			POP AJMP	YPOS XBLINAD	
02,2		, ,			HOME	YOUTHHU	; Build line address and return

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER *** Z-29 COMPUTER TERMINAL FIRMWARE VERS	(V1)-218 24-FEB-83 SION 1.03 ***	8 15:30:06 PAGE 127		
	;; AP1	IL - ANSI perform inser	rt line	
2 3 4	*AF	PIL* inserts the specie	fied Pn number of lines at the	
4 5 	; cur ;	rrent cursor position.		
6 8	; ; EN1	TRY none		
8 9 10	; ; EX1	IT none		,
10 11 12	; ; USE			
13				
14 0D7D 31 D1 15 0D7F 70 01 16 0D81 04	JN	ALL GNP Z APILO C A	; Get next parameter ; IF no parameter or O THEN ; bump to 1	
17	İNO			
18 0082 FA 19 0083 B1 47 20 0085 DA FC	APIL1: ACA	V INDX1, A ALL PIL	; Set up index ; REPEAT insert one line ; UNTIL count = 0	
20 OD85 DA FC 21 OD87 22	DUr RE		; UNTIL count = 0	

1; 1; 1;	7 8 9 0 1				7	P.DL	⇔erform delete line	· · · · · · · · · · · · · · · · · · ·
1; 1; 1; 1;	7 8 9 0 1				7			***************************************
1; 1; 1; 1;	7 8 9 0 1				ş	*PDL*		es of the current region up
1; 1; 1; 1;	7 8 9 0 1				;	one li	ine, clears the last line	e of the region, and moves
1; 1; 1; 1;	.2					the .cu	rsor.to.the.besinning.of	f the current line.
1; 1; 1; 1;	.2							
1; 1; 1; 1;	.2				; ;	ENTRY		
1; 1; 1; 1;	.2	••••••		*************	;	EXIT	none	······································
1; 1; 1;		•••••				USES	all	
					,			
1 6		co	10					
• •	.6 ODSA	CO	17	•••••	PDL:	P.VS.H PUS.H	TSCROLL	
. .	7.0D8C	CO	18			P.VSH	BFIX	······· <u>·</u> ····························
19	8 ODSE 9 OD90	E5 F5	12 F0			MOV VQV	A, YPOS	; Get current position
20	O OD92	B5	17	00		CJNE	A, TSCROLL, PDLA	; IF upper fixed THEN
21 22	11QUYS 2 OD97	49 04	ДЕ	•••••••••	P.DLA:	INC	Α	i.ust.return
23	3.0098	B5		Q. 4		CJNE	A. BEIX. PDL1.	i.IF.last.line.of.scnoll.THEN
	:4 OD9B :5.OD9D	71 \$0	96 D3		PDLO:	ACALL:	CLRLINE	; clear current line and return
20	:6			· · · · · · · · · · · · · · · · · · ·				
	7.009F 8 ODA1	40 B4	08	D1	PDL1:	JÇ CJNE	PDL2 A, #NUMLINES, PILRT1	; IF bottom fixed THEN
29	9 ODA4	75	F0 F2	18	•••••	MQV	B. #MAXLINE	; IF 25th line HEN ;clear 25th line
30	O ODA7	80	F2			SJMP	PDLO	; and return
32	2 ODA9	85	18	12	PDL2:	MOV	YPOS, BFIX	; Put cursor above fixed
	3 ODAC 4 ODAE	<u>15</u> 85	<u>12</u>	17	• • • • • • • • • • • • • • • • • • • •	DEC	YPOS TSCROLL, B	••••
35	5 ODB1	12 A1	0561 72	• ·		LÇALL	PLF	; Get saved current line
36	6 ODB4	A1	72		*****	AJMP	PILKET	; Keturn
	•••••		,	•••••	•••••			
	• • • • • • • • • • • • • • • • • • • •						·····	
••••••								
• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •					•••••		
					,		•••••	
	••••••	•••••••	*************	••••••	•••••	•••••		
		• • • • • • • • • • • • • • • • • • • •	•••••					
	• • • • • • • • • • • • • • • • • • • •							
	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	•••••		•••••	• • • • • • • • • • • • • • • • • • • •	••••	
								••••••

				RE VERSION 1.03 **			
<u>1</u>						ANSI perform delet	
3 4				, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	*APDL*	deletes the specif	ied Pn number of lines at and tion.
				;	+ollow:	ing the cursor posi	tion.
5					ENTRY	none	
g				······································			
					EXIT	none	
11					USES	A, B, DPTR, INDX1	
13				·····	···········ACALL···		; Get parameter
	ODB8	70	Dí 01	APDL:	JNZ	APDLO	; IF no parameter or O THEN
16 17	ODBA	გ4			INC	A	5 bump to 1
18	CODBB	FA		APDLO:	MÖV	"INDX1, A" PDL	; Set up index ; REPEAT delete one line
<u>19</u> 20	ODBC ODBE	B1 DA	88 FC	APDL1:	DUNŻ	INDXI, APDLI	; UNTIL count = 0
	opco	22			RET		
• • • • • •							
			• • • • • • • • • • • • • • • • • • • •				
 .							

))))))))

•

					a terral en la companya de la compa
9ROM; MICROBENCH	8051 CROSS ASSEM	BLER (V1)-218	24-FEB-83 15	5:30:06 PAGE 130	
*Z-28.QOMPUTERT	ERMINAL FIRMWARE	.XERSION 1.03.	***		
<u>1</u>	•••••			Perform insert chara	cter
2		5			
4	• • • • • • • • • • • • • • • • • • • •	***************************************	 Cursor	move.all.characters. one column to the r	that are at and to the right of the ight. Cursor position does not
5			chanae		***************************************
		, .			
8		•	ENTRY	none	
10	• • • • • • • • • • • • • • • • • • • •	***************************************	EXIT	none	
11			USES	all	
13		*		G. 1 1	
	50	PIC:	MOV	A, #NUMCHARS	* Calle within 10 and
16 ODC3 C3			CLR	C	; Calc number of moves to make
17 ODC4 95 18 ODC6 FA	11		SUBB MOV	A, XPOS	
19			MOV	INDX1, A	
20 ODC7 E5	11		MOV	A, XPOS	; Get current address
21 ODC9 85 22 ODCC 12	12F0		MOV LCALL	B, YPOS CLA	
23			LUMLL	ULH	
24 ODCF 78 25 ODD1 79	20 00		MOV	PTR1, #/ /	; Init value for 1st slot
26	······································	•••••	MQV	PTR2, #0	
. 27 ODD3 E0 28 ODD4 FC		PIC1:	MOVX	A, @DPTR	
29 ODD5 53	83 EF		MOV ANL	WORK, A DPH, #ATRANL	; Save lower character
30 ODD8 E0		••••••	MOVX	A, @DPTR	
31.0DD9FD 32			MOV	TEMP, A	; Save lower attribute
33 ODDA E9			MOV	A, PTR2	
34 ODDB FO 35 ODDC 43	00 40	-3	MOVX	@DPTR, A	; Write lower attribute
35 ODDC 43 36 ODDF E8	8310	•••••	ORL MOV	DPH, #CHRORL A, PTR1	
37 ODEO FO 38			MOVX	@DPTR, A	
	04		MOV	DTD1 HODKA	
40 ODE3 A9	04 05	•••••	MÖV	PTR1, WORKA PTR2, TEMPA	; Exchange character ; Exchange attribute
41 42 ODE5 A3			INC	DPTR	
43 ODE6 DA	EB		DJNZ	DPTR INDX1, PIC1	; Bump up ; UNTIL all done
44 ODES 22		•••••	RET	···;::::::::::::::::::::::::::::::::::	, GALIE ALL GOING
• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			
	• • • • • • • • • • • • • • • • • • • •				
	••••••	•••••	• • • • • • • • • • • • • • • • • • • •		
	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		
				* •	
	••••••	******************			
• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			
				•••••••••••	

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER *** Z-29 COMPUTER TERMINAL FIRMWARE VER	(V1)-218 24-FE SION 1.03 ***	B-83 15:	30:06 PAGE 131	
1	; ;	APIC -	ANSI perform insert chara	cter
1 2 3 4 5	; ; ;	*APIC* the lin	inserts the specified num	ber of characters into
6 7 8 9	; ;	ENTRY	none	
8 9	; 	EXIT	none	
10 11 12		USES	all	
13 14 ODE9 31 D1 15 ODEB 70 01	APIC:	ACALL JNZ INC	GNP APICO A	; Get parameter ; IF no parameter or O THEN ; bump to 1
17 18 ODEE FB 19 ODEF B1 C1 20 ODF1 DB FC	APICO: APIC1:	MOV ACALL DUNZ RET	INDX2, A	; Setup index ; REPEAT perform insert char ; UNTIL count = 0
21 ODF3 22				
······				
			*	
			• • • • • • • • • • • • • • • • • • • •	
		•••••••	······································	
			······································	
	· · · · · · · · · · · · · · · · · · ·		• • • • • • • • • • • • • • • • • • •	
·				
	······································			
		,		
			· · · · · · · · · · · · · · · · · · ·	

))))))

)

	1					PDC -	manfar 4.1.4	
	2				, , , , , , , , , , , , , , , , , , ,		perform delete char	acter
• • • • • • • • • • • • • • • • • • • •						*PDC*.	deletes the charact	er at the cursor position by moving
	5				;	the re	maining characters	on the line to the left one column the last column on the line. Cursor
	6				• • • • • • • • • • • • • • • • • • •	Positi	arnava.a.arava.av on does not change.	the tast tolown on the line. Corsor
	7							
	9				;	ENTOV		
	10			• • • • • • • • • • • • • • • • • • • •	· · · · · · ? · · · · · · · · · · · · ·	ENTRY	none	
	11					EXIT	none	
	12 13				; •	Here	7 1	
	14	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		USES	all	
	15 16 ODF4	<u></u>						
	16 ODF4 17 ODF6	74 C3	4F		PDC:	MOV	A, #MAXCHAR	; Calc number of moves
	18 ODF7	C3 95	11		• • • • • • • • • • • • • • • • • • • •	CLR SUBB	C A, XPOS	
	19.0DF9.	FA				MOV	INDX1, A	
	20 ODFA 21 ODFC	74 85	4F 12	EO		MOV	A, #MAXCHAR	; Get address at end of line
	22 ODFF	<u>12</u>	·····	F0		MOV LCALL	B, YPOS CLA	
	23 OEO2	E0				MOVX	A, @DPTR	
	24 0E03 25 0E04	F8 74	20			MOV	PTR1, A	; Save upper character
	26 0E06	·····/-7····	20	• • • • • • • • • • • • • • • • • • • •		MOV	A, #′ @DPTR, A	; Clear last character
	27 OE07	53 E0	83	EF		ANL	DPH, #ATRANL	, crear rast character
	28 OEOA 29 OEOB					MOVX	A, @DPTR	
	47 DEOB. 30 DEOC	F9	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		MOV CLR	PTR2, A	; Save upper attribute
	31 OEOD	FO				MOVX	@DPTR, A	; Clear last attribute
	32 OÉOÉ 33 OEOF	EA	00			MOV	A, INDX1	
	33 OEOF	60	20			JZ	PDC2	; Exit out if last column
	35 OE11	E5	82 FF		PDC1:	MOV	A, DPL	; Decrement DPTR without
	36 OE13 37 OE15	24		• • • • • • • • • • • • •		ÄĎĎ	A, #−1	; Usina a -CLR C-
•••••••	37 OE13 38 OE17	F5 	82 83			MOV MÓV	DPL, A A, DPH	; Instruction to save
	39 OE19	34	FF			ADDC	A, #-1	; Every bit of time
	40 0E1B	F5	83			MÓV	DPH, A	
	41 42 OE1D	E0	• • • • • • • • • • • • • • • • • • • •				A, @DPTR	
	3 0E1E	FD				MOV	TEMP, A	; Save lower attribute
	4 0E1F	43	83	iö		ORL	DPH, #CHRORL	
	15 0E22 16 0E23	<u>E0</u>	• • • • • • • • • • • • • •			MOVX	A, @DPTR	
4	17					MOV	WORK, A	; Save lower character
	8 0E24	E8				MÓV	A, PTR1	
	19 0E25 50 0E26	F0 53		<u>ė</u> F		MOVX	@DPTR, A	; Write character
	51 OE29	53 E9	ರಾವ	EF		ANL MOV	DPH, #ATRANL A, PTR2	
	52 0E2A	FÖ	• • • • • • • • • • • • • • • • • • • •			······································	edetr, A	; Write attribute
	53 54 OÉ2B''	×.						
	04 0E2B 55 0E2D	A8 A9	04			MOV MOV	PTR1, WORKA	; Exchange character
	6 0E2F	DA	05 E0		• • • • • • • • • • • • • • • • • • • •	DJNZ	PTR2, TEMPA INDX1, PDC1	; Exchange attribute ; UNTIL all done
	7 OE31	22			PDC2:	RET		, wille difficult

•

				APDC -	ANSI perform delete	character
1 2 3 4 5 6 7			,	*APDC*	deletes the specifi	ed number of characters from the
4 5			; ;	line.		
			**************************************	ENTRY	none	
<u>ś</u>				EXIT		
9 10					none	
11 12				USES	_all	
13 14 0632	3i	bi	APDC:	ACALL	GNP	; Get parameter
15 0E34 16 0E36	70 	01		JNZ INC	APDCO A	; IF no parameter or O THEN ; bump to 1
17	FB		APDCO:	MOV	INDX2, A	; Setup index
18 0E37 19 0E38	B1	F4 FC	APDC1:	ACALL	PDC INDX2, APDC1	; REPEAT perform delete char ; UNTIL count = 0
20 0E3A 21 0E3C	DB 22	FU		DJNZ RET	INDAZ, MEDCI	, ONTE COURT - O

•

2	• • • • • • • • • • • • • • • • • • • •	<u>i</u> .k	ZDSSBF	C.T.ZDS.set.baud.nate	2
. কু		; 	*ZDSSE	R*.sets.the.bayd_rat	e.af.the.WART.built.inta.the.8051.
5		; 			
		;	ENTRY	none	
8		*	EXIT	none	•••••••••••••••••••••••••••••••••••••••
10 11	• • • • • • • • • • • • • • • • • • • •	······································	USES	all	
12 13 OE3D 11	2A	ZDSSBR:	ACALL		
14				SETDISP	
15 OE3F 11 16 OE41 C3	2F		ACALL CLR	SETNORM C	
7 0E42 94 8 0E44 40	40		SUBB	A, .#:@::	; Subtract offset
9 OE46 B4		00	JC	SBRET	; Check for lower bounds
20 0E49 50	13	00 ZDS0:	ÇÜNE JNC	A#((N(±1@1), ZDS SBRET	Oi.Check.for.upper.bounds
2	• • • • • • • • • • • • • • • • • • • •	•••••••••••••••••••••••••••••••••••••••	IFC	 NE,SBR-\$	
3 4					T. FOLLOW
5 6					
27					
28 29		; ;	SBR -	set baud rate	
80 1		-			
2	• • • • • • • • • • • • • • • • • • • •	•••••••	*\$BR*.select.	selects a baud rate :	given the value of which rate to
3 4				•••••	
5 ś			ENTRY	(A) = which baud re	ate (013)
		,, ,	EXIT	none	
, ; · · · · · · · · · · · · · · · · · · ·			USES	. A. DPTR	
8 9	• • • • • • • • • • • • • • • • •				
8 9 0 1					
3 9 5 1 2 OÉ4B 90	0E5F	SBR:	MOV	DPTR, #BRTAB	; Point to baud rate table
3 9 0 1 1 2 OE4B 90 3 OE4E 93		SBR:	MOVC	A, @A+DPTR	; Point to baud rate table ; Get value from table
8 9 0 1 2 0E4B 90 3 0E4E 93 4 0E4F 60 5 0E51 E5	do	SBR:	MOVC JŽ	A, @A+DPTR SBRET	; Get value from table ; IF zero THEN exit
8 9 0 1 1 2 0E4B 90 3 0E4E 93 4 0E4F 60 5 0E51 F5 6 0E53 F5	OD 33 8D	SBR:	MOVC JZ MOV	A, @A+DPTR SBRET BAUDRATE, A	; Get value from table ; IF zero THEN exit ; Save baud rate
8 9 9 1 1 2 0E4B 90 3 0E4E 93 4 0E4F 60 5 0E51 F5 6 0E53 F5	OD 33 8D		MOVC JŽ	A, @A+DPTR SBRET BAUDRATE, A TH1, A	; Get value from table ; IF zero THEN exit ; Save baud rate ; Change baud rate
88 89 10 11 12 0E4B 90 13 0E4E 93 14 0E4F 60 15 0E51 F5 16 0E53 F5 17 0E55 C2 18 0E57 B4	0D 33 8D 9F 53	00	MOVC JŽ MOV MOV	A, @A+DPTR SBRET BAUDRATE, A TH1, A SMO	; Get value from table ; IF zero THEN exit ; Save baud rate
93 0E4E 93 44 0E4F 60 55 0E51 F5 66 0E53 F5 77 0E55 C2 8 0E57 B4 9 0E5A 50	0D 33 8D 9F 53		MOVC JZ MOV MOV CLR CJNE JNC	A, @A+DPTR SBRET BAUDRATE, A TH1, A SMO A,#83, SBRO SBRET	; Get value from table ; IF zero THEN exit ; Save baud rate ; Change baud rate ; Place UART into mode 1
38 39 10 11 12 0E4B 90 13 0E4E 93 14 0E4F 60 15 0E51 F5 16 0E53 F5 17 0E55 C2 18 0E57 B4 19 0E5A 50 10 0E5C D2	OD 33 8D	00 SBR0:	MOVC JŽ MOV MOV CLR CJNE JNC SETB	A, @A+DPTR SBRET BAUDRATE, A TH1, A SMO A,#83, SBRO	; Get value from table ; IF zero THEN exit ; Save baud rate ; Change baud rate ; Place UART into mode 1 ; IF baud <= 300 THEN
88 99 100 11 12 0E4B 90 13 0E4E 93 14 0E4F 60 15 0E51 F5 16 0E53 F5 17 0E55 C2 18 0E57 B4 19 0E5A 50	0D 33 8D 9F 53	00	MOVC JZ MOV MOV CLR CJNE JNC	A, @A+DPTR SBRET BAUDRATE, A TH1, A SMO A,#83, SBRO SBRET	; Get value from table ; IF zero THEN exit ; Save baud rate ; Change baud rate ; Place UART into mode 1
8 99 99 90 90 90 90 90 90 90 90 90 90 90	0D 33 8D 9F 53	00 SBR0:	MOVC JŽ MOV MOV CLR CJNE JNC SETB	A, @A+DPTR SBRET BAUDRATE, A TH1, A SMO A,#83, SBRO SBRET	; Get value from table ; IF zero THEN exit ; Save baud rate ; Change baud rate ; Place UART into mode 1 ; IF baud <= 300 THEN
8 99 99 90 90 90 90 90 90 90 90 90 90 90	0D 33 8D 9F 53	00 SBR0:	MOVC JŽ MOV MOV CLR CJNE JNC SETB	A, @A+DPTR SBRET BAUDRATE, A TH1, A SMO A,#83, SBRO SBRET	; Get value from table ; IF zero THEN exit ; Save baud rate ; Change baud rate ; Place UART into mode 1 ; IF baud <= 300 THEN

				BRTAB	.m.baud.mate	e.table		
	1 3 		; 	*BRTA	B*.contains.	the timer values.	for baud nates	
	 	OE5F	BRTAB	EQU				
	7		BRIAB		\$			
	"8'8E5F"" 9 0E60	01		DB	1	, , , , , , , , , , , , , , , , , , ,	75 baud	
• • • • • • • • • • • • • • • • • • • •	10.0E91	52 80		DB DB	82 128		110 baud	
	11 OE62	CO		DB	192	,	300 baud	
	12 0E63	EÓ.		DB	224	;	600 baud	
	13 0E64 14 0E65	F0 F5	• • • • • • • • • • • • • • • • • • • •	DB DB	240 245		1200 baud	
	15 0E66	00		DB	0		2000.baud.(not.avai.lable)	
	16 0E67	··řě		DB	248	;	2400 baud	
	17 0E68 18 0E69			DB DB	0		3600.baud.(not.available)	
	19 0E6A	00		DB DB	252 9	;	4800 baud (not available)	
•••••	20 0E6B	FE		DB	254	;	7200.baud.(not.available) 9600 baud	
	21 0E6C	. FF		pB	255		1.9.200. baud	
					,			
• • • • • • • • • • • • • • • • • • • •								
• • • • • • • • • • • • • • • • • • • •								
• • • • • • • • • • • • • • • • • • • •								
••••••								
• • • • • • • • • • • • • • • • • • • •					• • • • • • • • • • • • • • • • • • • •	,		
A								
		• • • • • • • • • • • • • • • • • • • •						

• • • • • • • • • • • • • • • • • • • •								
* * * * * * * * * * * * * * * * * * * *								

. 1	• • • • • • • • • • • • • • • • • • • •		PSOF	.evt.strina.in.oute	t FIFO
.3			*P\$0F*	Places the characte	r string pointed to by DPTR into haracter has bit seven set.
.ş					dideter has bit seven set.
. 7			ENTRY	(DPTR) = .s.tart.of.	strine
.9	• • • • • • • • • • • • • • • • • • • •	·····	EXIT	none	
10 11			USES	A, B, DPTR	
12 13	• • • • • • • • • • • • • • • • • • • •				
14 0E6D E4		PSOF:	CLR	A AALBETE	; REPEAT
15.0E6E93 16.0E6F 20	E7 OD		MOVC	A, @A+DPTR ACC.7, SOR	; set character ; IF bit 7 set THEN exit
17.9E72	82		PUSH	DPL	and any part are
17.9574Q 18 0E74 CO	94 83		PUSH	DPH	; save DPIR
18 0E74 CO 19.QE76D1	A&		AÇALL	CPR2	: place character on FIFO
12.0570P1 20 0E78 DO	#8 83			DPH	i place character on FIFO ; restore DPTR
21QE7AДQ	82				
22 0E7C A3			INC	DPTR	; bump pointer
23.QEZD8Q	EE		SJMP		; UNTIL done
4					
5.0E7F54	<i>T</i> F	SOR:	ANL	A., #7FH	* Mack off 7th hit
A APPARA AA					7 Mask off /th bit
26 0E81 80 	23		SJMP	CPR2	; Mask off 7th bit ; Place last char on FIFO
			SJMP	CPR2	
			SJMP	CPR2	
			SJMP	CPR2	
			SJMP	CPR2	
			SJMP	CPR2	
			SJMP	CPR2	
			SJMP	CPR2	
			SJMP	CPR2	
			SJMP	CPR2	
			SJMP	CPR2	
			SJMP	CPR2	
			SJMP	CPR2	
			SJMP	CPR2	
			SJMP	CPR2	

2						CPR -	cursor position repor	r <u>t</u>
3					;	*CPR*	outputs the cursor Po	osition in the format of the mode
4					• • • • • • • • • • • • • • • • • • •	curren	tly set.	
5								
7	<i>.</i>					ENTRY.	none	
8 9					; •	EXIT	none	
				•••••	······································			
11						USES	all	
13 14 OE8	93	30	43	11	CPR:	JNB MOV	HAZ, CPR1	; IF Hazeltine THEN
15 OE8		E5 B4	11	oo			A, XPOS A, #32, CPRO	; get -xpos-
17 OE8	3 B	50	02	-	CPRO:	JNC	CPROA	; IF value < 32 THEN
18 0E8	∃Ď	24	60			ADD	A, #96	; normalize
19 OE8	3F	D1	A4		CPROA:	ACALL.	CPR2	output value
20 OE9	91 20	E5	12 60			MOV ADD	A, YPOS A, #24	; get -ypos- ; normalize
21 0E9 22 0E9	7.4 75	.24 80	OF			SJMP	CPR2	; output value and return
23	<u></u>							; ELSE
24 OE9		90 D1	OEA9 6D		CPR1:	MOV ACALL	DPTR, #CPRM PSOF	; output ESC Y.
.25 OE9	78 70	E5	<u>99</u>			MOV	A, YPOS	; get line number
27 OE9		24	20			ADD	A,#.′′	; convert to ASCII
28 OE	40	D1	A6			ACALL	CPR2	; output it
.29.0E/ 30.0E/	92	E5 2 4	11			MOV	A, XPOS	; get column number ; convert to ASCII
30 OE4 31.QE4		2 4 92	20 022A		CPR2:	ADD JMP	A, #′ PCOF	; convert to ASCII ; output it and return
32								
33QE/	92	,1 B	P?		CPRM:	pa	ESC,((Y/:+80H)	
							,	
	• • • • • • •							

Z29ROM;	MICROBE	ENCH 805	51 CROSS	ASSEMBLE	R (V1)-218	24-FEB-83 15:	30:06 PAGE	138		
1				TOWARE.VE	RSION 1.03		identify te	rminal		
3 4 5					9 	DEC sof	identifies f ftware which would with a	n interrogate	as a DEC VT52 so that existies the console type will open	ng ate
				••••••	7 	ENŢŖŶ	none			
9 10 11 12						EXIT	none			
13 14						USES	all			
16 17	OEAE	. 90 C1	0EB0 6D 2F		IDT:	MOV AJMP	DPTR, #IDT PSOF		; Output string and return	
<u>19</u> 20		1B	 	CB	iotm:	DB	ESC, 777,	(′K′+80H)	; ESC / K	• • • • • • • • • • • • • • • • • • • •
21 22 23 24					· · · · · · · · · · · · · · · · · · ·	IDTT -	identify te	rminal type		
25 26					; ; ;	softwar	e can inter	the terminal rogate the c minal is run	as a Z-29 so that future onsole type and see what	
27 28 29 30				• • • • • • • • • • • • • • • • • • • •		ENTRY	none			• • • • • • • • • • • • • • • • • • • •
31 32 33			,			EXIT	none			• • • • • • • • • • • • • • • • • • • •
34 35 36						USES	all.			• • • • • • • • • • • • • • • • • • • •
38 39	OEB3 OEB5	11	2A 2F		IDTT:	ACALL ACALL	SETDISP SETNORM	• • • • • • • • • • • • • • • • • • • •	; Set dispatch and return ; Restore dispatch address	
41 42	OEBA OEBD	B4 90 C1	30 OEBF 6D	9B		CUNE MOV AUMP	A, #'O', Z DPTR, #IDT PSOF		; IF not 707 THEN return ; Output string and return	
43 44	ÖÉBF OEC2	1B BO	89	42*****	···ibttm:····		TESC, PIBY,	(k.0 s.±80H)	: ESC 1 B O	
	• • • • • • • • • • • • • • • • • • • •									
									••••••	
						•				
••••••										

ZZ9ROM; MICR *** Z-29 COM	ÖBENCH 80 PUTER TER	51 CROSS A MINAL FIRM	ASSEMBLER MWARE VER	(V1)-218 SION 1.03 *	24-FEB-83 15 ***	30:06 PAGE	139		
1			••••••			ANSI device	attributes		
3		•••••						as a DEC VT100 so that existing	9
4			• • • • • • • • • • • • • • • • • • • •	•	DEC so	ftware which	interrogates	as a DEC VT100 so that existing s the console type will operate	· · · · · · · · · · · · · · · · · · ·
<u>5</u>	· · · · · · · · · · · · · · · · · · ·			,	as it i	would with	VI100.		
<u>7</u>					ENTRY	none			
iö				, 					
10 11				;	EX17	none			
12			•••••	,	ÜŚĖŚ	ali			
13 14							•••••		
15 OEC3	31 70	D1 5É		ADA:	ACALL	GNPZÄŘ		; Get parameter ; IF no parameter or O THEN	
17 OEC7	90	OECC			MOV	DPTR, #AD	am .		
is deca		QD	.,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		AJMP	PSOF		; output string and return	
26 0ECC 0ECF	1B 31 E3	5B 3B	3F 32	ADAM:	DB	mesc, vt?i	(27, (7c7+80H)	
oeti2	E3								
•••••									
						· · · · · · · · · · · · · · · · · · ·			•••••
				• • • • • • • • • • • • • • • • • • • •					
•••••									

Z29ROM; MICRO *** Z-29 COMP	BENCH 80	51 CROSS ASS	SEMBLER (V1)-218 2 ARE VERSION 1.03.**	:4-FEB-83 15 *	:30:06 PAGE 140	
·	· · · · · · · · · · · · · · · · · · ·				.ZDSmode.tab.func	
2			; ;		.does.the.tab.func	
4 5			; :			
6			,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	ENTRY	none	
8 9			*	EXIT	none	
10 11		•••••	;	USES	all	······································
12	• • • • • • • • • • • • • • • • • • • •		7705-			· · · · · · · · · · · · · · · · ·
13.0ED3.		2A	ZTAB:		SETDISP	
15 OED5 16 OED7 17 OEDA	75	2FFO (OEDF	 D3	MOV	SETNORM B, #ZTABTL	
18 OEDD	01	36		VQM PMLA	DPTR#ZTABT STJMP	; Jump to any routine
19						
21 22	• • • • • • • • • • • • • • • • • • • •		•••••			······································
23 24	• • • • • • • • • • • • • • • • • • • •				ZDS.modetabtab.	
25 26		• • • • • • • • • • • • • • • • • • • •		*Z.TABT. functi	*containsvalide ons in ZDS mode.	anametens.for.the.set/clear.tab.
27 28	.,					
29 30	QEDF		ZTABT	EQU		
31.OEDF. 32 OEEO	0627			DB DW	HTC	; ESC . O ; Horizontal tab clear
	33	• • • • • • • • • • • • • • • • • • • •		DB	/3/	; ESC . 3
350EE3. 36	0598	••••••		WQ	CLRTABS	; Clear all tabs
37 OEE5 38 OEE6	38 0621			DB DW	8 <u>/</u> HTS	; ESC . 8 ; Horizontal tab set
39 40			ZTABTL	EQU	(\$-ZTABT)/3	; Table length
, ,						
						••••••
• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •				
	• • • • • • • • • • • • • • • • • • • •					
			• • • • • • • • • • • • • • • • • • • •			

Z29ROM; *** Z-2	MICROBE 9 COMPUT	NCH 8051 ER TERMI	CROSS AS NAL FIRM	SEMBLER NARE VERS	(V1)-218 24-F ION 1.03 ***	EB-83 15	:30:06 PAGE 14	41				
1					5 5	SCP -	save cursor Po	osition				
3	3				;		saves current		tion.			
4	}				;							
	7		• • • • • • • • • • • • • • • • • • • •		,	ENTRY	none					
					· • · · · · · · · · · · · · · · · · · ·	EXIT	none				•• •••••	
íč					÷ · · · · · · · · · · · · · · · · · · ·	···USES···	none	•••••		• • • • • • • • • • • • • • • • • • • •		
12	2	 os		 25	SCP:	MOV	YYPOS. YPOS		; Save values	•••••••	••••	
12	OEE8	. 85 	.11	.35 .36		MOV MOV	XXPOS, XPOS XYPOS, YPOS					
15	OEEE					RET					• • • • • • • • • • • • • • • • • • • •	
17 18	, 3										•••••	
17 18 19 20 21 22 25 25 26 27 26 27 30 30	? 				;;	ASCP -	ANSI save cui	rsor position	n		• • • • • • • • • • • • • • • • • • • •	
21	l 2				• • • • • • • • • • • • • • • • • • •				nd attributes.			
23	3 1				,	,					•••••	
25	5 					ENTRY	none					
	? 7 				<u>.</u>	EXIT	none					
29	» 9 				, , ,	USES	all					
30	i 2. oeee 3. oeee						ĠŃĖ			,.,		
32	2 OEEF 3 OEF1 4 OEF3	31 40 22	D1 F5		ASCP:	ACALL JC RET	GNP SCP		; Get paramete ; IF no parame	r 1f any ter THEN ok	••••••	
34	1 OEF3	22				RET						
							· · · · · · · · · · · · · · · · · · ·					
***************************************								• • • • • • • • • • • • • • • • • • • •			••••••	
***************************************											•••••	
			• • • • • • • • • • • • • • • • • • • •								•• • • • • • • • • • • • • • • • • • • •	
											• • • • • • • • • • • • • • • • • • • •	
											••••••	
***************************************											•••••	
											• • • • • • • • • • • • • • • • • • • •	

))))))))

Z29R01 *** 7-	M; MICROL	BENCH 805	1 CROSS	ASSEMBLER	R (V1)-218 2 RSION.1.03.**	24-FEB-83 15	:30:06 PAGE 142			
	. 1						.unsaye cursor posit	tion		
	34						returns to Position		the use of *SCP*.	
	.5 6 7	• • • • • • • • • • • • • • • • • • • •	••••••			ENTRY	none			
	8				···	EXIT	none	•••••		
1	10 11 12				,	USES	all			
1	12 13 OEF4 14 OEF7	85 E5	35 36	11	USCP:	MOV	XPOS, XXPOS A,XYPOS	; Re	store values	
1	15 OEF9 16 OEFC	<u>B4</u>	18 15	03 02		CUNE UNB	A, #MAXLINE, USCP1 L25EN, USCP2		on 25th line THEN IF not enabled THEN exit	
i	17 OEFF 18 OFO:	F5 2i	12 F4		USCP1: USCP2:	MOV AJMP	YPOS, A XBLINAD	: FI:	E set position ouild address and return	
	19 20 21		• • • • • • • • • • • • • • • • • • • •							
	21 22 23 24	•••••			••••••••••••••••••••••••••••••••••••••	AUSCP -	- ANSI unsave cursor	······································		•••••
	24 25 26				, , , , , , , , , , , , , , , , , , ,		* returns to positio		th the use of *SPCP*.	
	26 27 28					······································				
2	29 36 · · · · · · · ·				, , , , , , , , , , , , , , , , , , ,	ENTRY	none none			
	31 32					USES	all			
	33 34 · · · · · · · · · · · · · · · · · · ·									
	35 0F03 36 0F05 *** 37 0F07	31 40 22	ED		AUSCP:	ACALL JC RET	GNP USCP		: parameter if any no parameters THEN ok	
							•••••			•••••••••••••••••••••••••••••••••••••••
		,								
• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •				••••••			• • • • • • • • • • • • • • • • • • • •	•••••	
						,		•••••		

1		5 5	ZATR -	ZDS select attrib	utes
2 3		; ;	*ZATR*	selects which att	ributes are to be turned on or off
4 5	• • • • • • • • • • • • • • • • • • • •	9	usina	the configuration	of the lower 5 bits of the character.
6					
7 8		,	ENTRY	none	
9 			EXIT	none	
) 1		;	USES	all	
1 2					
3 4 OFOS 11	2A	ZATR:	ACALL	SETDISP	; Set dispatch and return
5 6 OFOA 11	2 ;		ACÁLL	SETNORM	; Restore dispatch address
7 OFOC C3 3 OFOD 94	<u></u> 1		CLR	C A, #202	
3 OFOD 94	30		SUBB	A, #/0/	TITTE I TO THE TOTAL TOTAL
9 OFOF 40 0 OF11 B4	<u>14</u>		JC CJNE	ZAR A, #32, ZATRO	; IF below range THEN exit
O OF11 B4 1 OF14 50	0F	o ZATRO:	JNC	ZAR	; IF above range THEN exit ; Get reverse video
2 OF16 13			RRC	A	; Get reverse video
3. 9F17 92	08		<u>Moy</u>		; Get blinking
4 OF19 13	00		RRC MOV	A BLINKA, C	
5.0F1A92 6.0F1C 13	<u>9</u> 4		RRC	Α	; Get half intensity
7 OF1D 92 8 OF1F 13	OB		MOV	HALFIA, C	
8 OF1F 13			RRC	A	; Get underline
9 OF20 92 O OF22 13	99		MOV RRC	UNDLNA, C.	; Get alternate char set
0 0F22 13 1.0F2392	oç.		MOV	ALTCHARA, C	, det diterinate and
2 OF25 22		ZAR:	RET		
ş					
4					
5 6				,	
			XATR	exit all attribut	es
8		;			withthe or that many and active
?		?	*X+;1,1;5;	. Turus . orr. a.u. a.u	ributes so that none are active.
o .1		, , , , , , , , , , , , , , , , , , ,			
2		,	ENTRY	none	
ą					
4 5		; •	EXIT	none	
쿠 6			USES	none	
·7					
8	, <u>.</u>	o vato.	ANII	ATTOIDHTEC #AEA	OH ; Mask off all attributes
9.9F2453 0 0F29 22	21F	oXATR:	ANL RET	ATTRIBUTES, #OFG	41
0 0FZ9 ZZ			114		
				• • • • • • • • • • • • • • • • • • • •	

	. <u>1</u>			EHALF - enter half intensity mode
	.3		;	*EHALF* places the terminal into half intensity mode.
	4		,	
• • • • • • • • • • • • • • • • • • • •	6		\$ 	ENTRY none
	. <u>í</u> 8		······ ·	EXIT none
	.9 io	• • • • • • • • • • • • • • • • • • • •	······································	USES none
	l 1 l 2	•••••		
	l3 OF2A D l4 OF2C 2	2OB	EHALF:	SETB HALFIA ; Set half intensity attribute RET
	15 16	- -		, nc1
	17			
	18 19		; ;	EUNDL – enter underline mode
2	20 21		;	*EUNDL* places the terminal into underline mode.
	21 22 23		**************************************	Acousts Fraces the terminal into underline mode.
· · · · · · · · · · · · · · · · · · ·	23 24			ENTRY none
 2 2	25 26	•••••		EXIT none
·····	?7 ?8	,		USES none
· · · · · · · · · · · · · · · · · · ·	:9 :::::::::::::::::::::::::::::::::::			
	81 OF2D D 82 OF2F 23	209	EUNDL:	SETB UNDLNA ; Set underline attribute RET
3	:3 :4	• • • • • • • • • • • • • • • • • • • •		REI
	5 			
	16 17 18		; ;	EBLNK – enter blink mode
3	9		*	
4	0	•••••	• • • • • • • • • • • • • • • • • • •	*EBLNK* places the terminal into blink mode
	<u>1</u>	• • • • • • • • • • • • • • • • • • • •		ENTRY none
4		• • • • • • • • • • • • • • • • • • • •		EXIT none
4				USES none
4	7 8			
4	9 OF30 D2 0 OF32 22	9OA	EBLNK:	SETB BLINKA ; Set blink attribute
		.		RET
				······

Z29ROM; MICROBENCH 8051 CROSS ASSEMBL *** Z-29 COMPUTER TERMINAL FIRMWARE V	ER (V1)-218 24-F ERSION 1.03 ***	FEB-83 15:30:06 PAGE 145	
1	3 5	ERVM – enter reverse video mode	
3	; ;	*ERVM* places the terminal into reverse video mode.	
4	*		
		ENTRY none	
7 8		EXIT none	
9 10		USES none	
11			
12 13 OF33 D2 O8	ERVM:	SETB RVVA ; Set reverse video attribute RET	
14 0F35 22		RET	
15 16	•••••		
1718			
		XRVM - exit reverse video mode	
21 22		*XRVM* removes the terminal from reverse video mode.	
22 23	; 		
24	; :	ENTRY none	
25 26	· · · · · · · · · · · · · · · · · · ·	EXIT none	
27. 28		USES none	
29 30			
31 OF36 C2 08 32 OF38 22	XRVM:	CLR RVVA ; Clear reverse video attribute.	

	.1		EGM – enter araphics	mode	• • • • • • • • • • • • • • • • • • • •
	3 4	, , , , , , , , , , , , , , , , , , ,	*EGM* Places termina	l:.into.graphics.mode.	•••••
• • • • • • • • • • • • • • • • • • • •	5	· · · · · · · · · · · · · · · · · · ·	ENTRY none		
•••••	./ 8 9		EXIT none	······································	
1	0 1	······································	USES none		
1	2 3 0F39 D2 OD 4 0F3B 22	EGM:	SETB GRPH RET	; Set graphics mode	
	5		RET		
	.7 8				•••••
	9		XGM — exit graphic m		•••••
2 2	11 12		*XGM* exits the term	inal out of graphic character mode.	
2	3		ENTRY none		
2	.6 7 8	······ • ····•	EXIT none	······	
2	8 9 	,	USES none		•••••
	1 0F3C C2 OD 2 0F3E 22	XGM:	CLR GRPH RÉT	; Clear graphic mode	
					• • • • • • • • • • • • • • • • • • • •
•••••					

			; ;	GODES	– GO designator – G1 designator	
			;			
				*GODES	* and *G1DES* assis	n a graphic character set to be
				associ	ated with itself.	If the character set is not found
			; •		o action is taken. ent character sets	The values returned for the are as follows:
• • • • • • • • • •	• • • • • • • • • • •			,		
				Charac	ter ! Internal ed ! value and mea	
			;	rece1v	ed : value and mea	
		• • • • • • • • • • • • • • • • • • • •		Α	0 = United 9	tates (should be United Kingdom)
				В	0 = United 9	tates tates with alternate graphic characters
			; •	O 1	1 = United 8	tates with alternate graphic characters e character set
	• • • • • • • • • • • • • • • • • • • •			·····	: 3 = Alternat	e with alternate graphic characters
			.			
			5	CALTON		
					none	
			i	EXIT	none	
			5	uoco	11	
		,		UŞEŞ	A U	
OF3F	11	2A	GODES:	ACALL	SETDISP	; Set dispatch and return
OF41	11	2F		ACALL	SETNORM	; Restore dispatch address
OF43	ç2	p5		CLR	F0	; Flas for GODES
0F45	80	08		SJMP	GDES	; Do the stuff
	• • • • • • • • • • • • • • • • • • • •					
	,					
05.47			ma meno-	ACALL	CETTICO	. Cat dispatch and natura
. QF.4.7	1.1	2A	G1DES:		8¢1/hY5L	; Set dispatch and return
0F49	1.1	2F		AÇALL.	SETNORMFO	Restore dispatch address
OF4B	D2	D5		SETB	F0	; Flag for G1DES
	80	QQ		SJMP		; Do.the stuff
0545	T1.4		03 GDES:	CJNE	A, #'A', GDES1	; IF char = 'A' THEN
OF4F .QF52	B4 E4	41	OS GDESO:	CUNE CLR	A, #'A', GDESI	; value of zero
OF53	80	10		SJMP	GDES3	
·						• TE -k ZDZ TUCN
OF55	B4 \$0	42 F8	02 GDES1:	CJNE SJMP	A, #′B′, GDES2 GDES0	; IF char = 'B' THEN i
Yr.ƏR	₽₩				'A. M. M. M. A	
. QF5A	çş		GDES2:	CLR	ç	
OF5B	94	30		SUBB	A, #101 GDESR	; Subtract offset
OF5D OF5F	40 B4	1B 03	00	UÇ CJNE	A, #3, GDES2A	; IF out of bounds THEN return
. 0F3F 9F62	50		GDE\$2A:	JNC	GDESR	; IF out of bounds THEN return
OF64	04			INC	Α	: Bump into range

	1.0F.65 2 0F68	29 53	H.P	99	GDES3:	JB ANL	FO, GDESSA		IF for GO THEN clear old value
	3 0F6B	42	1F	rv		ORL	GSET, #111100 GSET, A	1008 ;	clear old value
	4 OF6D	30	12	OB		UNB	GSEL, GOPICK		Put in new value IF selected THEN pick it
	5.0FZ9	22				RET		7	
	6	r	4						
	/ <u>95./1</u> 8 OF74	53 C4	1F	QF	GDES3A:	ANL	GSET, #000011	11B ; [ELSE clear old value Place into proper nibble
		42	15			SWAP ORL	A COET A	5	Place into Proper nibble
1	0 0F77	20	1F	07	• • • • • • • • • • • • • • • • • • • •	JB	GSET, A GSEL, G1PICK		Put in new yalue IF selected THEN pick it
1	1 2 OF7A							,	
1:	2 OF7A	22			GDESR:	RET		,	Return
• • • • • • • • • • • • • • • • • • • •									
	• • • • • • • • • • • • •					• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	
							• • • • • • • • • • • • • • • • • • • •		
						• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		
					• • • • • • • • • • • • • • • • • • • •				
							• • • • • • • • • • • • • • • • • • • •		
• • • • • • • • • • • • • • • • • • • •									
					• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
				• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •		

			5 5	GOPICK GIPICK	 pick GO graphic se pick G1 graphic se 	<u>t</u>
			7 7 7 7	······********************************	K* and *G1PICK* place	the terminal into the mode set for the particular graphic
			; ;	ENTRY	none	
			,	EXIT	none	
			\$	USES	all	
OF7B OF7D	C2 E5 80	12 1F 05	GOPICK:	CLR MOV SJMP	GSEL A, GSET GPICK	; Clear select flas ; Get graphic set choosen
OF7F OF81	D2	12	G1PICK:	SETB	GSEL	; Set select flag
.0F83 0F85	E5 C4	1F		MOY	A, GSET A	; Get araphic set choosen
OF86	53	21 4 F	GPICK:	ANL	ATTRIBUTES, #01001	11B ; Clean different modes out
OF89	 54	 OF		ANL	A, #00001111B	; Mask to set number of set
OF8B	90	0F92		MOV MOVC	DPTR, #GPICKT A, @A+DPTR	; Get address of table ; Get value from table
OFSE OFSF OFS1	93 42 22	21		ORL RET	ATTRIBUTES, A	; Set as indicated by table
? }						
; 5				GPICKT	– graphic pick tabl	 e
				GPICk	(T contains values t	nat will be ORed into the
				as inc	oute pyte to set ine. Hicated by the graphi	different character set modes - mode defined above.
	QE92.		GRICKI	EQU	 \$	
QE.92			••••	DB	00000000B	; O - norm char
oF.93				DB	10100000B	; 1 - norm char and alt graphic set.
:	10			DB	00010000B	; 2 - alt char set
				DB	10110000B	; 3 — alt char and graphic set

	SISTALL STEEL	3333664 12. 12. 12.	MWARE.YERSION.1.03.*	: ቶ ቶ			
				SBLR -	set blink rate		
			, , , , , , , , , , , , , , , , , , ,	*SBLR*	sets the blink rate	e in increments of 1/30 second. The	
4 ≕			,	defaul	t value is 32/30 sed	cond. If a value of 255 is received	
6 	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • •	····· ·	tḥeṇ ạ is rec	ull blinking data is reived then all blink	suppressed. If a value of O	
7						vins data is continuously on.	
			;	ENTRY	none		
10							
11				EXIT	none		
1314			, , ,	USES	all		
14 15							
16 OF96	31	Di	SBLR:	ACALL	GNP		· · · · · ·
17 OF98 18 OF9A	50 74	<u>02</u> 20		JNC MÖV	SB1	; IF no parameter THEN	
19				MOV	A, #32	; set to default value	
20 OF9C 21 OF9E	F5 60	37	SB1:	MOV	BLNKRATE, A	; Set blink rate	• • • • •
22 OFA5	B4	06 FF		JZ CUNE	SB2 A, #255, SBRT	; IF not zero THEN ; IF max value THEN	
23 OFA3 24 OFA4	B3	oi		SETB	C	; flas off and	
25 25 26 OFA6		01		SUMP	SB3	; set output	
26 OFA6 27	çэ	•••••	SB2:	CLR	ċ	; ELSE IF zero THEN flag on	
28 OFA7	<u>92</u> 	oi	SB3:	MOV	BLINKF, C	; Set output of blink	
29 OFA9 30 OFAB	<u>. E</u> 5	20			A, XLATCH DPTR, #MLATCH		
31 OFAE 32	FO			MOVX	@DPTR, A	; Set latch	
32 33 OFAF	75	38	00 SBRT:	MOV	BLNKCNT, #0		
34 OFB2	75 22			·····RET	DEMICIALLY #O	; Zero count and return	
			• • • • • • • • • • • • • • • • • • • •				
	,	• • • • • • • • • • • • • • • • • • • •					• • • • •
• • • • • • • • • • • • • • • • • • • •							
				· · · · · · · · · · · · · · · · · · ·			
			• • • • • • • • • • • • • • • • • • • •				
							
• • • • • • • • • • • • • • • • • • • •							
						•••••	

C C C C C C C C C

2			,,	EKI -	enable keyboard in	Put
3			7 7			ich disables input from the keyboard.
5			,			
			**************************************	ENTRY	none	
·····			······································	EXIT	none	
				ÜSÉS	all	
					KBDISF	
13 OFB3 14 OFB5	C2 74	1F 89	EKI:	CLR MÓV	A, #KBC_EK	; Enable keyboard
15 OFB7 16	61	2A		AJMP	ХРСКВ	; Output and return
17 18						
9 0 1 2			, , , , , , , , , , , , , , , , , , , ,		disable keyboard i	nout
<u>.</u> 1			, , , , , , , , , , , , , , , , , , ,			
2 3			"	*UK1*	sets the flas whic	h disables input from the keyboard.
23 24 25 26			;	ENTRY	none	
7			,	EXIT		
7 7 8 9 0			; •		all	
Z 0					3!.\	
1 2 OFB9	pż	iF	DKI:	SETB	KBDISF	
3 OFBB 4 OFBD	74 61	88 2A		MOV AJMP	A, #KBC_DK XPCKB	; Disable keyboard ; Output and return
	•••••					
				,		

Z29F ***	ROM; Z-29	MICROBE COMPUI	NCH 805	51 CROSS 91NAL.FI	ASSEMBLE RMWARE.YE	R (V1)-218 RSION 1.03	3 24-FEB	-83 15:	30:06 PAGE 1	52	• • • • • • • • • • • • • • • • • • • •				
	1							КВСРЦК.	m. kemboandcs	aes.lock.co	utine				
• • • • • • • • • • • • • • • • • • • •	2 3 4 5					; <u>;</u> ;		*KBCPLK locked		e,kemboand.	into.and	.o.uto.ftl	ne.caps	••••••	
	<u>6</u>	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		; ;			none	,	• • • • • • • • • • • • • • • • • • • •				
••••••	8 9					**************************************			 					•••••	
	10 11 12					; ;			A. DPTR					• • • • • • • • • • • • • • • • • • • •	
	13	• • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		ATAT						• • • • • • • • • • • • • • • • • • • •	
	14	OFBF OFC1 OFC3	B2 74 20	35 84 35		KBCPLK:		CPL MOV JB	CAPLOCKF A, #KBC_EL CAPLOCKF, KO		; Toggi	e caps l	cked fla	 .9	
	.17.	OFC6 OFC8	74 71 61	85 2A		KC1:		MOV ACALL	A,#KBC_XL XPCKB						
	.19.0	OFCA		23				AJMP	XR2_DISP		, outpo Displ	it comman .av.statu:	o to keyb S.Jing.&.	oard netwnn.	
			• • • • • • • • • • • • • • • • • • • •												
							••••••••••••••••••••••••••••••••••••••							• • • • • • • • • • • • • • • • • • • •	
• • • • • • • • • • •				•••••			• • • • • • • • • • • • • • • • • • • •						• • • • • • • • • • • • • • • • • • • •		
• • • • • • • • • • • • • • • • • • • •					• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •								
• • • • • • • • • • • • • • • • • • • •														• • • • • • • • • • • • • • • • • • • •	 ••••••
• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •												• • • • • • • • • • • • • • • • • • • •	
• · · · · · · · · · · ·								• • • • • • • • • • • • • • • • • • • •				• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	
	• • • • • •					• • • • • • • • • • • • • • • • • • • •		· · · · · · · · · · · · · · · ·				• • • • • • • • • • • • • • • • • • • •			
			• • • • • • • • • • •	······											
• • • • • • • • • • • • • • • • • • • •				•••••										• • • • • • • • • • • • • • • • • • • •	
	•••••	•••••		•••••									• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
• • • • • • • • • • • • • • • • • • • •				•••••										• • • • • • • • • • • • • • • • • • • •	
•••••	• • • • • • • •			•••••											
		•••••													
				•••••			••••••		•••••	• • • • • • • • • • • • • • • • • • • •					
*********														• • • • • • • • • • • • • • • • • • • •	
														••••••	

• • • • • • • • • • • • • • • • • • • •									•••••	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •		

1		;;	KBBREAK	– keyboard break key		
34		,			break key on the keyboard has should then hold the serial	• • • • •
5				line low for a length		
7		; ;				
8		• • • • • • • • • • • • • • • • • • •	ENTRY	none		
9 10			EXIT	none		
12 13		ţ	USES	all		
14						• • • • •
15 OFCC C2 16 OFCE 7A	B1 00	KBBREAK:	CLR MOV	TXD INDX1, #0	; Set break	• • • • •
17 OFDO 7B	50		MOV	INDX2, #80		
18 OFD2 12	02B6	BR1:	LCALL	FCKB	; REPEAT look at keyboard	
19 OFD5 40 20 OFD7 B4	05 89 02		JC CUNE	BR2 A, #KB_BREAK, BR2	; IF another break THEN	• • • • •
21 OFDA8Q	F.Q	BR2:	SUMP	KBBREAK	; start asain	
22 OFDC DA23 OFDEDB	F4 F2	BR2:	DÜNZ	INDX1, BR1 INDX2, BR1	; UNTIL done ; UNTIL all done	
24 OFEO D2	B1	• • • • • • • • • • • • • • • • • • • •	SETB	TXD	; Back to normal	
25.0FE222			RET			
				•••••		

)

Z29	PROM; MICRO ≱.Z-29.COMPU	BENCH 80! UTER TERI	51 CROSS ASSE MINAL FIRMWAR	MBLER (V1)-218 24 E.VERSION 1.03.***	-FEB-83 15			
	1					OLLkeyboard."NO.SCRO		••••••
,	34					ROLL* indicates that the ard has been depressed.	e "NO SCROLL" key on the	
	5 6 7					none		
	8 9			; 				
	12 13			KBSCROLL:				
	14 OFE3 15 OFE6 16 OFE8	20 74 	36 OA 13 11 O2		JB MOV JNB	HSMODEF, KBSCR2 A, #XOFF UXOFFSENT, KBSCR1	; IF not hold screen mode THEN ; start with XOFF	
		74	11		MOV	A, #XON	; IF user XOFF last sent THEN ; send an XON now instead	
	20			KB\$CR1:		PCQF		
	21 OFFQ 22 OFF2 23 OFF4	C2 05 30	1E	KB\$CR2:	ÇLR INC JNB	HSLINE	; ELSE allow screen to rvn ; bump line count ; IF shift down THEN	
	24 OFF7 25 OFF9	E5 Ç3	18		MOV ÇLR	A, BFIX	; calculate number ; between pages	
	26 OFFA 27 OFFC 28 OFFE	95 F5 22	17 1E	KSR:	SUBB MOV RET	A, TSCROLL HSLINE, A		
			••••••					
	••••••	,	•••••••					• • • • • • • • • • • • • • • • • • • •
			***************************************			•••••		•••••
								••••••••
						•••••		
	• • • • • • • • • • • • • • • • • • • •				• • • • • • • • • • • • • • • • • • • •			
	• • • • • • • • • • • • • • • • • • • •				• • • • • • • • • • • • • • • • • • • •			
		•••••		• • • • • • • • • • • • • • • • • • • •	•••••			
	••••••	• • • • • • • • • • • • • • • • • • • •						
	••••••	•••••	••••••		•••••	•••••		• • • • • • • • • • • • • • • • • • • •
								•••••
			• • • • • • • • • • • • • • • • • • • •					
								•••••

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE 155 *** Z-29 COMPUTER TERMINAL FIRMWARE VERSION 1.03 ***
1

)

)

)

)

)

<u>1</u>			. 5. 5	ROM #2 vectors
2 3			; ;	
4 5			<u> </u>	These locations are provided to facilitate future modifications without a change to the masked processor. Each location is
<u>.</u>		• • • • • • • • • • • • • • • • • • • •	?	accessed with a -CALL- and control is sained back with a -RET
				Initialization of terminal.
·····iŏ·····	1000			ORG R2_IN
	0002	10AF		JMP ALLINIT
13 14			···	
15 16			, ,	Soft initialization of terminal.
17 100	1003 02	10E4		ORG R2_IN2 JMP SOFTINIT
18				
19 20 21		• • • • • • • • • • • • • • • • • • • •	 5	Start DMA routine.
22	1006			ORG R2_STARTDMA
23 100 24	06 02	11DC		LJMP STARTDMA
25 26				Stop DMA routine.
27 28			,, ,,	
29 100	1009 19 02	i1FD		ORG R2_STOPDMA LJMP STOPDMA
30 31 32				
32 33	•••••		*	Fill screen routine.
34 35 100				ORG R2_FILL
36	C 02	1262		LJMP STFILL
37 38		• • • • • • • • • • • • • • • • • • • •	. :	Interrupt request.
39 40	100F	• • • • • • • • • • • • • • • • • • • •	. 5	ORG R2_IRQ
41 100 42	F 02	1329		LJMP IRQ
43	• • • • • • • • • • • • • • • • • • • •			
44 45		,	*	Serial interrupt.
46 47 101	1012 2 02	12CB		ORG R2_SERIAL
48	 		••••••	LJMP SERIAL
50 				Transmit character from serial port.
51 52	1015			ORG R2_XMT
53 101 54	5 02	1398		LJMP TRANSMIT
55 55			,	Send XON to host
56 57	1018		;	ORG R2_XON

C)

1018	02	1305		LJMP	DOXON
<u> </u>					
) 5			; •	Initia	lize the cursor format.
	101B	• • • • • • • • • • • • • • • • • • • •		ÖRĞ	R2_ICUR
101B	02	117B		LJMP	SETCF
· ?					
) l			; ;	Initia	lize the keyboard
2	101E	1105		ORG LJMP	R2_IKB
ŀ	02	1175	•••••	LJMP	INITKB
5 				Setue	mode vector
7 	,		.		
3 2 1021	1021 02	1417		ORG LJMP	R2_SETUP SETUP
)					······································
l 2		• • • • • • • • • • • • • • • • • • • •	· · · · · · · · · · · · · · · · · · ·	Backer	ound loop hook.
} }	1024			ORG	R2_BACK
. 1924	22			RET	NZ_DHON
) ,					
3			; ;	Keyboa	rd input hook.
?)	1027	• • • • • • • • • • • • • • • • • • • •		ORG	R2_KY1
1927	22			RET	
2 3					
} 5			; ;	Keyboa	rd output hook.
5	102A			ORG	R2_KY2
7192A 3		•••••		RET	
?)				Ferano	parsing hook.
			,		
2 31920	102D 22			ORG RET	R2_ESC
		•••••			
5 5		•••••		Transm	it character.
7 3	1030			ORG	R2_TC
	Q2	108B			SXMTC
) 					
2			;	Transm	it line.
3	1033			ORG	R2_TL
	02	1.DA\$		LJMP	SXMTL

Z29ROM; MICROBENCH 8051	CROSS ASSEMBLER (V1)-218 24-FE	EB-83 15:30:06 PAGE 158
*** Z-29 COMPUTER TERMIN	AL FIRMWARE VERSION 1.03 ***	
······ 2		Transmit page.
3 1036 4 1036 02 5	1 DAC	ORG R2_TP LUMP XMTP
6	•••••	
		Transmit 25th line.
	1BC6	ORG R2_T25L LJMP XMT25
11 12 13		Print page.
13 14	•••••••••••••••••••••••••••••••••••••••	
	1002	ORG R2_PRNT LUMP PRINT
17 18		
<u>19</u> 20		Display clock.
21 103F 22 103F 02	161F	ORG R2_DCLK LJMP DCLK
23 24	••••	
25 25	5	Display status.
27 1042	,	ORG R2_DISP LJMP DISPSTAT
28 1042 - 02	1E69	LJMP DISPSTAT
		······································
••••••	•••••	
•••••		
•••••		
•••••		
		······
		······

729ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE 159

1				; ;	INIT -	initialize terminal	
2				,			
3 					*!N!!*.	sets up internal KAM tables and initialize	from the ROM and EAROM, reconfigures
4 5				;	screen	(ables and initialize	s Processor brites.
6				, , , , , , , , , , , , , , , , , , ,			
7					ENTRY	none	
8 o				;	EXIT	none	
9 9 0							
1					USES	all	
2 3							
4 1045	75	po	00	INIT:	MOV	PSW, #BANKO	; Select bank O
5							; Load constants for
6 1048	78	oc			MOV	PTR1, #ISTORE	
7 104A 8 104D	90 7A	1FD1	• • • • • • • • • • •		MOV MOV	DPTR, #RAMCONST INDX1, #25	; initialization from ROM
9	7 11	17				INDXI7 WES	
0 104F	11	A7			ACALL	INMOV	; Move data
1						BTD4 AVVDOO	
2 1051 3 1053	78 70	35 0B			MOV MOV	PTR1, #XXPOS INDX1, #11	; Load constants for
9 4		YR					
5.1055	1.1	A7			AÇALL	INMOV	; Move data
6				_		C EAGGM	
7 8					Kecall	from EAROM	
9 9 1057	C2	03		,	CLR	EAROMRCALL	
0 1059	C2 12	03 1E18			CALL	WLATCH	; Recall from EAROM
11Q5Ç	p2	Q3			SETB	EAROMRCALL	
2 105E	12	1E18			CALL	WLATCH	; By togaling bit
3 4 1061	78	25		• • • • • • • • • • • • • • • • • • • •	 MOV	PTR1, #25H	; Load constants for
5.1063	9.Q	7000			MQV	DPTR, #EAROM	; Loading RAM from EAROM
6 1066	7A	10			MOV	INDX1, #16	
7				IN3:	MOVX	A, @DPTR	; REPEAT get upper nibble
8 1068 2.1069	E0 54	OF		103:	ANL	A#QFH	, REFERRI Set OPPER HIDDIE
7. AV97 0 106B	9.7 C4	y)			SWAP	Α	· · · · · · · · · · · · · · · · · · ·
1. 1.96C	FD				MQV	TEMP, A	; save_upper_nibble
2 106D	A3				INC	DPTR	; bump EAROM pointer
3.106E	<u>E</u> Q				MQVX	A, @DPTR A, #OFH	i get lower nibble
4 106F 5 1071	54 4D	OF			ANL ORL	A, #OFH A, TEMP	; mask data ; combine upper and lower
로 1974 6 1072	7 µ		• • • • • • • • • • • • • • • • • • • •		MOV	@PTR1, A	; place into internal RAM
7.1973	АЗ				INC	DPTR	; bump EAROM pointer
8 1074	08				INC	PTR1	; bump internal RAM pointer
? 1 975	PA	F1			DJŅZ	INDX1,IN3	
0 1 1077	co	ec			CLR	TRO	; Turn off timer O
1.1077 2.1079	<u>92</u> 75	8C 89	23	• • • • • • • • • • • • • • • • • • • •	MOV	TMOD, #ITMOD	; Initialize values
3.107C	75	98	5A		MQV	SCON, #ISCON	; Place UART into mode 1
4						0 00000075	. 0.4 1
5 1 A 7 C	<u>E</u> 5 F5	33 8D			MQV MOV	A, BAUDRATE	
5.107F 6.1081	c=					11142 61	, THILIDITER COURT LAILE

			200 200 CO. C. C. C. S. S. C.	3***		•••••••••••••••••••••••••••••••••••••••
11.084	50	92	IN3B:	JNC	IN3A	; IF baud <= 300 THEN
2 1088	D2	9F		SETB	SMO	; place UART into mode 3
31.08A	D2	\$E	IN3A:	SETB	TR1	. Turn on baud rate timer
4 5.108Ç	A2	30			C DOTES	
6 108E	92	99 02		MOV	C.PRTF2	; Initialize port direction
7. 1090	12				WLATCH	
8 1093	A2	ЗD		MOV	C, ACHR2	; Get alternate char set flas
91.975	?2	QC	• • • • • • • • • • • • • • • • • • • •	MOV	ALTCHARA, C	; Place in attributes
10 11 1097	30	43	02	15.175	1107 7110	
12 109A	<u>30</u>	7 9 D4		JNB ACALL	HAZ, IN4 HAZINIT	; IF Hazeltine mode THEN ; initialize for Hazeltine
13				HUHLL	HEINII	
14 109C	12	1F22	IN4:	CALL	NVRCHK	; Checksum the NVRAM
15, 109F	B5	34	01	CUNE	A, NVRSUM, INS	; IF same THEN
16 10A2	22			RET		; return
17.19A3		40	10 IN5:	ORL	ERRORS, #00010000B	; ELSE error #4
18 10A6	22			RET		
<u>19</u> 20		• • • • • • • • • • • • • • • • • • • •	•••••	• • • • • • • • • • • • • • • • • • • •	•••••	
21			;			
22			;		rom ROM to RAM using DP	TR, PTR1, and INDX1
23						
24 25. 10A7	FΔ		INMOV:	ÇĻR	^	• DEDEAT
26 10A8	 7		AM19X1	MOVC	A A, @A+DPTR	; REPEAT ; load constant
2710A9	F6			MOV		, road constant , Place into internal RAM
28 10AA	АЗ	• • • • • • • • • • • • • • • •		INC	DPTR	; bump ROM pointer
29 10AB 30 10AC	08					
				INC		
30 10AC	DA	F9		IŅC	PTR1 INDX1, INMOV	
30 10AC 31 19AE		F9		INC DUNZ RET	PTR1 INDX1, INMOV	
30 10AC 31 10AE	DA	F9		DUNZ	PTR1 INDX1, INMOV	; bump RAM pointer ; UNTIL done
30 10AC 31 10AE	DA	F9		DUNZ	PTR1 INDX1, INMOV	; bump RAM pointer ; UNTIL done
30 10AC 31 10AE	DA	F9		DUNZ	PTR1 INDX1, INMOV	; bump RAM pointer ; UNTIL done
30 10AC 31 10AE	DA	F9		DUNZ	PTR1 INDX1, INMOV	; bump RAM pointer ; UNTIL done
30 10AC 31 10AE	DA	F9		DUNZ	PTR1 INDX1, INMOV	; bump RAM pointer ; UNTIL done
30 10AC 31 10AE	DA	F9		DUNZ	PTR1 INDX1, INMOV	; bump RAM pointer ; UNTIL done
30 10AC 31 10AE	DA	F9		DUNZ	PTR1 INDX1, INMOV	; bump RAM pointer ; UNTIL done
30 10AC 31 10AE	DA	F9		DUNZ	PTR1 INDX1, INMOV	; bump RAM pointer ; UNTIL done
30 10AC 31 10AE	DA	F9		DUNZ	PTR1 INDX1, INMOV	; bump RAM pointer ; UNTIL done
30 10AC 31 10AE	DA	F9		DUNZ	PTR1 INDX1, INMOV	; bump RAM pointer ; UNTIL done
30 10AC 31 10AE	DA	F9		DUNZ	PTR1 INDX1, INMOV	; bump RAM pointer ; UNTIL done
30 10AC 31 10AE	DA	F9		DUNZ	PTR1 INDX1, INMOV	; bump RAM pointer ; UNTIL done
30 10AC 31 10AE	DA	F9		DUNZ	PTR1 INDX1, INMOV	; bump RAM pointer ; UNTIL done
30 10AC 31 10AE	DA	F9		DUNZ	PTR1 INDX1, INMOV	; bump RAM pointer ; UNTIL done
30 10AC 31 10AE	DA	F9		DUNZ	PTR1 INDX1, INMOV	; bump RAM pointer ; UNTIL done
31 10AE	DA			DJNZ RET	PTR1 INDX1, INMOV	; bump RAM pointer ; UNTIL done
31 10AE	DA			DJNZ RET	PTR1 INDX1, INMOV	; bump RAM pointer ; UNTIL done
31.10AE	DA 22			DJNZ RET	PTR1 INDX1, INMOV	; bump RAM pointer ; UNTIL done
31 10AE	DA 22			DJNZ RET	PTR1 INDX1, INMOV	; bump RAM pointer ; UNTIL done
31 10AE	DA 22			DJNZ RET	PTR1 INDX1, INMOV	; bump RAM pointer ; UNTIL done

					ALLINI	T — initialize all of	terminal
				, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	*ALLIN	IT* initializes every	thing including the CRTC.
					ENTRY	none	
					EXIT	none	
				····· ,	USES	all	
 10AF	C2	AF		ALLINIT:	CLR	EA	; Disable interrupts
10B1	75	90	CF		MOV	P1, #11001111B	; Initialize port #1
10B4 10B7	75 12	40 1ÉC4	00		MOV CALL	ERRORS, #0 KBCHK	; Initialize errors here ; Check keyboard first (timing)
10BA	<u>i</u> 1	·····45			ACALL	init	; Initialize everythins
TOBC	E4 78				ĊĽŔ MOV	A PTR1, #LORDER	; Setup to initialize
10BF	78	47 19			MOV	INDX1, #NUMLINES	; the -LORDER- table entries
1001 1002	F6 08			ALLI1:	MOV INC	@PTR1, A PTR1	; REPEAT place entry into table ; bump RAM pointer
1003 1004	O4 DA	FB			INC DUNZ	A INDX1, ALLI1	; bump count ; UNTIL done
1006	75	B8	03		MOV	IP, #IPINIT	; Interrupt priorities
1009 1008	31 12	1E 1ED9			ACALL.	INITORT ROMCHK	; Start CRT controller ; Check ROM
10CE	12	1EF1			CALL	RAMCHK	; Check RAM
1.0D1	75	A8	95		MOV	IE, #IEINIT	; Enable interrupts
1004 1007	<u>12</u> 75		18		CALL	CLRS B, #MAXLINE	; Clear screen
10DA 10DD	12 D2	0B96 24			CALL SETB CALL	CLRLINE VSPF ERROR	; Clear 25th line ; Flas screen to be turned on ; Display any errors
10DF 10E2	21	1F2C 9E			AJMP	INITKB	; Initialize keyboard & return

••••••			;		UI. – soft initializ	
				ACCETT	NIT's initializate th	
			**************************************	curren	t settings such as	e terminal but does not destroy the caps lock, on/off line, and time on
· · · · · · · · · · · · · · · · · · ·	• • • • • • • • • • • • • • • • • • • •		······	thec).	ock	
				ENTRY		
					•••••	
	• • • • • • • • • • •	· · · · · · · · · · · · · · · · · · ·	; ;		none	
			5		all	
	•••••	•••••		•••••	•••••	
10E4 10E6	CO	3A	SOFTINIT:	PUSH	TCNT	; Save clock information
1.QE8	co	3C	•••••	PUSH	TMIN	
10EA 10EC	CO A2					
10EE	92	ΕO		MOV	ACC.O, C	; Save online flag
10F2	.#4 92	E1	•••••	MOV	ACC.1, C	; Save caps locked flag
10F.4	. <u>CO</u>	EQ		P.USH	ACC	
1.0F.6	7.5	90	CB	MOV	P1, #11001011B	
1.0F.2	. 1.1	45		ACALL	•	; Initialize everything
	DO.	F0				
10FD	A2			MOV		
1101	A2	EO		MOV	CAPLOCKF, C	
1103	.92	2Ę		MOV	ONLINE, C	; Restore online flas ; Restore clock information
1107	DO	30 30		POP POP	THOUR TMIN	; Restore clock information
1109 110B	DO DO	3B		POP	TSEC	
			•••••			
1.10D	.31	7B		ACALL.	SETCF	; Set cursor format
110F	.12	OBAC		CALL	CLRS	; Clear screen ; IF 25th line is off THEN
1115	.75	57 F0	06 18	JB MQV	B, #MAXLINE	
1118	02	OB96		JMP	CLRLINE	; clear 25th line
111B	02	1E69	SOFTI1:	JMP	DISPSTAT	; ELSE display status info
	10E8 10EA 10EC 10EE 10F0 10F2 10F4 10F6 10F8 10FB 10FB 110F 1103 1105 1107 1109 1108 110B	10E6 C0 10E8 C0 10EA C0 10EA C0 10EC A2 10EE 92 10F0 A2 10F2 92 10F4 C0 10F6 75 10FB D0 10FB A2 110F A2 1101 A2 1103 92 1105 D0 1107 D0 1108 D0 1109 D0 110B D0 110D 31 110F 12 1112 20 1115 75 11118 02	10E6 CO 3B 10E8 CQ 3C 10EA CO 3D 10EA CO 3D 10EC 92 EO 10FO A2 35 10F2 92 E1 10F4 CO EO 10F6 75 90 10F9 11 45 10FB DO EO 10FB A2 E1 10FF 92 35 1101 A2 E0 1103 92 E 1100 3D 1107 DO 3C 1109 DO 3B 1109 DO 3A 110D 31 7B 110D 31 7B 110F 12 0BAC 1112 20 37 1115 75 FO 1118 02 0B96	10E4 CO 3A SOFTINIT: 10E6 CO 3B 10E8 CO 3C 10EA CO 3D 10EC A2 2E 10EE 92 EO 10F2 92 E1 10F4 CO EO 10F6 75 90 CB 10FB DO EO 10FB A2 E1 10FF 92 35 1101 A2 E0 1103 92 2E 1105 DO 3D 1107 DO 3B 1109 DO 3B 1109 DO 3A 110D 31 7B 110F 12 OBAC 1115 75 FO 18 1118 O2 OB96	10E6 CO 3B PUSH 10E8 CO 3C PUSH 10EA CO 3D PUSH 10EC A2 2E MOV 10EE 92 EO MOV 10FO A2 35 MOV 10F2 92 E1 MOV 10F4 CO EO PUSH 10F6 75 9O CB MOV 10F9 11 45 ACALL 10FB DO EO POP 10FB DO EO MOV 10FF 92 35 MOV 1101 A2 E1 MOV 1103 92 2E MOV 1105 DO 3D POP 1107 DO 3C POP 1109 DO 3B POP 1109 DO 3A POP 110D 31 7B <	STATE STAT

(

				5	INITCR	T — initialize CRT co	ntroler
				;	*INITC	RT* does everything n	eeded to set the CRT controler
				· · · · · · · · · · · · · · · · · · ·	up and	running Properly.	
					ENTRY	none	
				,	EXIT	none	
				*	USES	A, DPTR	
111É	co	A8		INITCRT:	PÚSH	IE	; Save interrupt setup
1120	C2	AF			CLR	EA	; Disable interrupts
1122		09	02		MOV	DMLNA, #2	; Start at first line to DMA
1125	75 75	ÖB	02 A0		MOV	DMADRL, #LOW (DMAME	M+160)
1128	75	OA	60		MOV	DMADRH, #HIGH (DMAM	EM+160)
112B	D2	97			SETB	CLKRUN	; Let the clock run
112D	31	52			ACALL	SETCRT	; Set CRT controler
						ITO	; Int O transition activated
112F 1131	D2 53	88 88	D1		SETB	TCON, #11010001B	; Clear IT1, IEO, IE1, TFO
	90	2001			MOV	DPTR, #DCMDREG	
1134		2001 E0			MOV MOV		
1137 1139	74 F0				MOVX	A, #CSET @DPTR, A	; Preset counters
113A	c2	97			CLR	CLKRUN	; Turn off clock
1130	74	20			MOV	A, #CSTRT	
113C	74 F0	20			MOVX	@DPTR, A	; Start CRT controler
			,		SETB	CLKRUN	; Start clock
1.13F	Þ2	97	• • • • • • • • • • • • • • • • • • • •		эьль		
	7.4	AO			MOV	A, #CINT @DPTR, A	; Enable VRTC interrupts
1143	FO	14					, Englis vivio Interrupts
!					IFC	NE, DSTAREG-DCMDREG	IDPEC/ and different
					ERROR ENDC	, politice and pur	IDREG′ are different
		· · · · · · · · · · · · · · · · · · ·			MOV	DPTR, #DSTAREG	
• •		•		¥	MOV	DEIRS #DOINKED	
1144	ΕO				MOVX	A, @DPTR	; Get status register
1145	44 64	B3			ORL XRL	A, #10110011B A, #00001000B	; Fill in dont cares ; Tossle proper positions
1149	F.4					A	
114A	60 43	03 40	04		JZ ORL	IC1 ERRORS, #00000100B.	; IF not zero THENflas error #2
1.14Ç	†. 9	7 9	Y	••••••			•
114F	¤0 22	88		IC1:	P.QP	IE	

	<u>.</u>					SETCRI	set CRT controler	(reset 8276)
	2 3 4				; 	*SETCR	T* initializes the C	RT.controler.screen.parameter.bytes.
	5 8				, ,	ENTRY	none	
	7 					EXIT	none	
 10					,	UŚEŚ	A, DPTR	
	2	90	2001		SETCRT:	MOV	DPTR, #DSTAREG	••••••
1 '	3 1152 4 1155 5	90 E0				MOVX	A, @DPTR	; Clear status resister
1 · · · · · · · · · · · · · · · · · · ·	5 7	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		IFC ERROR	NE,DCMDREG-DSTAREG ;'DCMDREG' and 'DS'	TAREG′ are different
19	9					ENDC	<	
20) l 2 1156	• • • • • • • • • • • • •			;	MOV	DPTR, #DCMDREG	
21 <u>21</u>	2 1156 3 1158 1	7 4 F0				MOVX MOV	A, #CRESET @DPTR, A	: Reset and stop display
24 25 33	1159 1150	90 74	2000 4F			MOV	DPTR, #DDATREG	
20	7 115E 3	74 F0	46			MOV MOVX	A, #SCRN1 @DPTR, A	; Output screen parameter bytes
29	, , 115F , 1161	<u>74</u> 	98 39			MOV JB	A, #SCRN2A FREQ, SC1	; Check for 50/60 Hz
3:	1164	74 F0	DC		SC1:	MOV	A, #SCRN2B @DPTR, A	7 Greek 101 30700 112
33	1167	74	99			MÖV	A, #SCRN3	
35 37	1169	FO				MOVX	@DPTR, A	
38	116A 116C	<u>74</u>	4E	02		JB	A, #SCRN4A FREQ, SC2	Check for 50/60 Hz
40	116F 1171 1173	<u>74</u>	4F 38		SC2:	MOV	A, #SCRN4B C, CRNBK	; Get cursor no blink flas
42	1175	92 A2 92	E5 32 E4	• • • • • • • • • • • • • • • • • • • •		MOV MÖV MOV	ACC.5, C C, CRUL ACC.4, C	; Get cursor underline flag
4.2	1179 117A	… န်စ် 22				MOVX RET	edetr, A	
							• • • • • • • • • • • • • • • • • • • •	
••••••			• • • • • • • • • • • • • • • • • • • •					
•••••			• • • • • • • • • • • • • • • • • • • •					

	M; MICROB	ENCH 805	1 CROSS A	SSEMBLER WARE VERS	(V1)-218 ION 1.03	24-FEB-83 1	5:30:06 PAGE	165					
					5.5		– set curso	r format					
					;		F* initializa		ontroler f	or the cur	sor format		
					; ;	indic	ated.						
				• • • • • • • • • • • • • • • • • • • •	;	ENTRY	none						
	7 8 9 10				; ;		none						
	10				;		A, DPTR						
	11 12			• • • • • • • • • • • • • • • • • • • •	f								
	13 14 117B	31	52		SETCF:	ACALL	SETCRT		; Initi	alize scre	en registe	rs	,
	15 16 117D	<u>9</u> 0	2001		•••••	MOV MOV	DPTR, #DC	MDREG					••••••
	17 1180 18 1182	74 F0	20	• • • • • • • • • • • • • • • • • • • •		MOVX	A, #CSTRT @DPTR, A		; Start	CRT contr	oler		
	19 20 1183	22		• • • • • • • • • • • • • • • • • • • •		RET							
		• • • • • • • • • • • • •				•••••••							••••••
													••••••••
		· · · · · · · · · · · · · · · · · · ·											
,													

)

Z29ROM; MICRO *** Z-29 COMP	BENCH 80: VJER JERI	MINAL FIRMWAR	E. YERSION 1.03 **	* *		
1					update cursor posit	tion
3 4 5			5	disabl	es cursor if data is 3	turson address to the CRTC and > 600 baud or display of cursor
6 7		• • • • • • • • • • • • • • • • • • • •		isd.is.	abled.	••••••
8 9 10			* *	ENTRY	none	
10 11 .12				EXIT	none	
13				USES	A, DPTR	••••••
15 1184 16 1187	90 74	2001 80	CURSOR:	MOV MOV	DPTR, #DCMDREG A, #CCRSR	
17 1189 18	F0			MOVX	@DPTR, A	; Load cursor Position
19 118A 20 118D 21 118E	90 E5 F0	2000		MOV MOV	DPTR, #DDATREG A, XPOS @DPTR, A	
21 118F 22 23 1190		12		MOVX MOV	A, YPOS	; Load cursor X-position
24 1192 25 1195 26 1198	E5 36 30	12 1A 03 14 02 1A		JNB	ENBLCUR, CR1 DX, CR2	; IF cursor display disabled OR ; IF char royd since last time
26 1198 27 119A 28	74 F0	1A 	CR1: CR2:	MOV MOVX	A, #NUMLINES + 1 @DPTR, A	; put in non-existina region ; ELSE put cursor Y-position
29 119B 30 119D	C2 22	14		CLR RÉT	DX	; and clear -DX-
		,				
				•••••		
				•••••		

i			;;	INITKB	- initialize keybo	ard
2			,	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		
3				*INITK	B* resets the keybo the current moment	ard to whatever settings the terminal
† 5			,	nas at	the carrent moment	•
5			• • • • • • • • • • • • • • • • • • •			
7				ENTRY	none	
3			3	CVIT		
?			·····	ÉXIT	none	
1			;	USES	all	
2						
3 4 119E			INITKB:	IFC	NE,KBC_XL - KBC_E	
, 112E			INTIND.	ERROR	This code will	
				ENDC		V. F T. V. (1
7.119E	74 20	84 35		MOV	A, #KBC_EL	
3 11AO	20	35	01	JB INC	CAPLOCKF, IKB1	; IF not caps locked THEN
7 11A3 7 11A4	<u>04</u>	c8	IKB1:	ACALL	ннннн	; output caps unlocked ; (save bytes this way)
i				,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		, (3470 5100 6113 641)
2				IFC	NE,KBC_ONLN - KBC	
3				ERROR	:::This.code.will.	not workí
4 511.A6	74	94		MOY	A, #KBC_OFFLN	
5 11A8	30	8A 2E	01	JNB	ONLINE, IKB2	; IF on line THEN
7.11AB	04			INC	A	; output on line command
3 11AC	31	C8	IKB2:	ACALL	IKB6	
? >				IFC	NE,KBC_EK - KBC_I	K = 1
					ifThis.code.will.	
2				ENDC		
311 <u>0</u> E 4 11B0	74 20	88 1F		MQV,	A, #KBC_DK KBDISF, IKB3	; IF keyboard disabled THEN
4 1180 51183		1F 	01	INÇ	A	; ir keyboard disabled them ; enable keyboard
v• A.Q.Q 5 11B4	31	cs	IKB3:	ACALL	IKB6	
7						
3				IFC	NE,KBC_DC - KBC_E	
?)				ENDC	i::This.code.will	ΠQ\$, ΨQCK
, 11.1.136	74	82		MQY	A, #KBC_EC	
2 11B8	20	33	01	JB	CLKF, IKB4	; IF key click disabled THEN
311BB	04		IKB4:	INC	AIKB6	; output no key click
4 11BC 5	31	C8	1KB4:	HUHLL	TVDO	
₹ 5				IFC	NE,KBC_DR - KBC_E	R - 1
7					i/This.code.will	not_work(
3		0.4		ENDC	A AKDO CO	
7.11BE 0 11CO	7. 4 20	84 34	01	MQ.V JB	A₁#K₿Ç⊥ĒŖ REPF, IKB5	; IF auto repeat disabled THEN
1.11C3	04	U-7	W.A.	INC	A INDS	; no auto repeat
2 1104	31	C8	IKB5:	ACALL	IKB6	
3	74				A #250 T5	The second secon
4 1106 5 1108	7 4 02	80 026D	IKB6:	VOM PML	A, #KBC_ID PCKB	; Have it send it's ID number

.....

1.						TERMIN	ITterminal.initialia	ation
3					;			rminal for ZDS and ANSI
4					5	emulat:	ion mode. This places	the terminal in whatever mode
a.	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • •	.		rminal.must.be.in.to.en	nulate those modes.
7 8	• • • • • • • • • • •					ENTRY	none	······
? 10				• • • • • • • • • • • • • • • • • • • •				
.11.					, , , , , , , , , , , , , , , , , , ,	EXIT	none	
12 .13.					;	USES	all	
14 15	.11CB	. 53	21	FO	TERMINIT:	ANL	ATTRIBUTES, #OFOH	; Clear attributes
16	11CE	D2	1B			SETB	ERM	; Set erasure mode
	.11D0 11D2	D2 80	25 			SETB SUMP	GATM HIR	; Set suarded area xmt mode ; Clear protected and return
19. 20								
21. 22 23.								
24		•,••••••	,	• • • • • • • • • • • • • • • • • • • •				ation
.25. 26	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •				*HAZIN	IT*.initializes.the.ter	minal for the Hazeltine 1500 the terminal in whatever mode
.27								ulate the Hazeltine 1500.
28 .29					; ;			
30 31.					*	ENTRY	none	
32					·	EXIT	none	•••••••
33. 34		.,				USES	all	
,35 36				• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	,	
	.11D4 .11D6	. D2 D2	OB 18		.HAZINIT:	SETB SETB	HALFIA ERM	; Starts in background ; Reset erasure mode
39.	.1.1 D.S	.75	3F		.HIR:	MQV	PFIELD, #0	; No protected fields
40	11DB	22				RET		
							• • • • • • • • • • • • • • • • • • • •	
			• • • • • • • • • • • • • • • • • • • •		•••••	••••••	••••••	•••••
					• • • • • • • • • • • • • • • • • • • •			
• • • •		• • • • • • • • • • • • • • • • • • • •						
							••••	
	· · · · · · · · · · · · · · · · · · ·							
• • • •								
	•••••	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •				

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE 169 *** Z-29 COMPUTER TERMINAL FIRMWARE VERSION 1.03 *** STARTDMA - start psuedo DMA transfer *STARTDMA* can only be interrupted by the stop DMA routine. It is initiated by pushing the address to start DMA onto the stack and executing a -RET-. The hardware takes over and executes an instruction that is hard wired to the data bus and initiates reads from the video memory to the CRT controler. ENTRY none EXIT none 12 13 USES none 15 16 17 11DC 18 11DE PUSH ACC ĊŌ. EO PUSH . B.... CQ. FQ. 19 11E0 20 ; Select interrupts for DMA. MOV. IE, #IEDMA .21.11E2. ; Set type for DMA transfer В4 SETB DMATYPE 22 11E5 D2 A, DMLNA MOV 24 11E7 E5 09 ; IF DMA for line #0 THEN SD1 05 ŅZ .25.11E9. . 70. IF no suppression THEN JNB VSPF, SD1 26 11EB 30 24 02 ; always turn video onVSP..... . 9.2. CLR. . C2...27..11EE..... 28 ; Push DMA start address. PUSH 29 11F0 ; onto the stack PUSH DMADRH OA 30 11F2 CO ; Initialize timer values MOV. THO: #QEEH .gç..... .FF.31..11F.4. . 75. . ; For 80 characters MOV TLO, #OD6H 32 11F7 75 8A D6 ; Start timer running... ; Start esuedo DMA SETR. TRQ. 33.11FA.....D2... ... 8Ç..... RETI 34 11FC

····· !					STOPD	1A stop psuedo DMA trac	nsfer
3				;	*STUDI	IMO* is the other half of	f the DMA transfer routing.
4				**************************************	It st	ops the timer, cleans the	stack, sets -DMATYPE- to it's
5					normai	l value, and enables into	errupts. If it was filling memory
6. 7				;	then :	it finishes up that other	wise it continues and calculates
····/	• • • • • • • •	• • • • • • • • • • •			the ne	ext_line_number,_address,	and indexes for DMA.
9							
10				;	ENTRY	none	•••••
					EXIT	 none	
13				, ,	EXII	none	
14				*	USES	none	•••••••••••••••••••••••••••••••••••••••
5			• • • • • • • • • • • • • • • • • • • •				
17 11FD	.02 75	sc		STOPDMA:	CLR	TRO	; Stop timer
	75	A8	95		MOV	IE, #IEINIT	; Reset interrupt enables
. <u>19</u> 20 1202	 20	B4	20		JB	DMA TARRET PARA	
	D2	B4	20		SETB	DMATYPE, DM1 DMATYPE	; IF filling memory THEN
	DO	83			POP	DPH	; set -DMATYPE- to normal ; set last address from
	.DO	82			POP	DPL	; the stack
	AD	FO			MOV	TEMP, B	; get remainder
	.BD 32	00	01		CJNE	TEMP, #0, FILL	; IF nothing left THEN
	74	20		FILL:	RETI MOV	A #/ /	; return
28 1213	53	20 83	CF	!	ANL	A, #′ ′ DPH, #DMAANL	; ELSE set up and
	43	83	10		ORL	DPH, #CHRORL	; mask into char memory
	A3			FL1:	INC	DPTR	REPEAT bump position
	F0 53	83	EF		MOVX	@DPTR, A	fill with spaces
33 121E	50 F0				ANL MOVX	DPH, #ATRANL @DPTR, A	mask into attribute mem
34 121F	43	83	10		ORL	DPH, #CHRORL	; clear attributes ; mask into char memory
35 1222	DD	F5			DJNZ	TEMP, FL1	UNTIL index1 = 0
36 1224 37	32				RETI		; Return
		• • • • • • • • • • • •				• • • • • • • • • • • • • • • • • • • •	
	75 DO	<u>D0</u>	08	DM1:	MOV	PSW, #BANK1	; ELSE select bank #1
					POP	ACC	; clean off stack
.41.122A	DO	E0			POP	ACC	
	09				TNC	TIMI N	
44	7			• • • • • • • • • • • • • • • • • • • •	INC	DMLN	; bump current DMA line
	20	39 1D	13 02		JB	FREQ, DM1A	; IF 50 Hz THEN
	B9		02		CUNE	DMLN, #29, DM1B	; IF line 29 THEN
	80 B9	13 19			SUMP	DM2	; wrap around
	89 40	19 10	00	DM1B: DM1C:	CUNE JC	DMLN, #NUMLINES, DM1C	; ELSE IF line <= 25 THEN
	 B9	10	öö	! !!!!!:	CUNE	DM3 DMLN,#NUMLINES+1, DM1D	.; do usual stuff
	40	10		DM1D:	JC	DM4	
	D2	92		••••••	SETB	VSP	; ELSE IF line >= 26 THEN ; begin blanking
53 1241 :	80	18			SUMP	DM4	
	B9	19	00	DM1A:	CUNE	DMLN,#NUMLINES, DM1E	* Charle for one
	40	19		DM1E:	JC	DM3	; Check for wrap around ; IF line > 25 THEN

Z29R0M;	MICROB	ENCH 8	51 CROSS	ASSEMBLE	R (V1)-218	24-FEB-83 1	5:30:06 PAGE 1	71	• • • • • • • • • • • • • • • • • • • •				
*** Z-2	a cowbo	TER TE	KWINAL FIR	MMAKE VE	RSION 1.03 *	+ * *		• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •			
1	124A	74	47		DM3:	MOV	A, #LORDER		: FISE e	et base	nf table		
3	124C	<u>29</u> F8	······································			ADD	A. DMLN		; add	in offse	t		
	124D 124E	F8 E6				MOV MOV	DMPTR, A A, @DMPTR		; move ; set	it into value fr	pointer om table		
6													• • • • • • • • • • • • • • • • • • • •
	124F 1252	75 A4	F0	50		MOV MUL	B, #NUMCHAR	S	.;set.	up for m t	ultiply		
9	1253	F5 E5	oB Fo			MOV	DMADRL, A		; save	low ord	er byte		
	1255 1257	E5	F0 60			MOV ADD	A, B A, #HIGH (D	MAMEM)	: add	high ord			
12	1259	24 F5	<u>60</u>			MÖV	DMADRH, A	!!!!!!!!!!!!	; save	piap or	der byte		• • • • • • • • • • • • • • • • • • • •
13	1258	ро	Fö		DM4:	PÖP	<u>B</u>		Restor	e regist	ers and r	 turn	
15	.125D	DO	EO DO			POP	ACC 'PSW						
16	125F 1261	D0 32	DO			POP RETI	'PSW -						
		Υ		• • • • • • • • • • • • •					• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	
•••••													
• • • • • • • • • • • • • • • • • • • •													
										• • • • • • • • • • • • • • • • • • • •			
•••••			• • • • • • • • • • • • • • • • • • • •										
							• • • • • • • • • • • • • • • • • • • •						
										• • • • • • • • • • • • • • • • • • • •			
			• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •					
										• • • • • • • • • • • • • • • • • • • •			
			• • • • • • • • • • • • • • • • • • • •				,		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • •	
	• • • • • • • • • • •							• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
				• • • • • • • • • • • • • • • • • • • •					• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			
										• • • • • • • • • • • • • • • • • • • •			
									• • • • • • • • • • • • • • • • • • •				

))))))))

)

1				STF.IL	L.H.start.filling.wit	th spaces
4 3			; ;			.of.characters.and.location.in.
4				scree	n memory and fills th	em with 20H and places the
え ^	• • • • • • • • • • • • • • • • •			attri	bute siyen into attri	bute memory. It takes into
7				accou:	nt the mode that ERM	is in.
8 9 :			, 	ENTRY	(A) = number of ch	aracters less than 127
O 1			;		(B) = attribute to	· write
‡ 2					(DPTR) = character	memory address to start from
3 4				EXIT	none	
5			, ,	USES	A, B, DPTR, WORK2,	TEMP, INDX2, TIMER O
6 7 8 1262 FB						
8 1262 FB			STFILL:	MOV	INDX2, A	; Save count
9. 1263AF. O	F9			MQY	WORK2, B	
11265BF. 2	ΩΟ	3D		ÇUŅĒ	WORK2, #0, SFB	
3126830	1B	04	• • • • • • • • • • • • • • • • • • • •	JNB	ERM, SFA	
4 5.126B	3F					
6 126D 70				MOV JNZ	A, PFIELD SFB	
7 3 126F 53	 83					, precidi
3 126F 53	83 83	EF 20	SFA:	ANL	DPH, #ATRANL	; convert to DMA memory
9.127243 0.1275EB			• • • • • • • • • • • • • • • • • • • •	ORL	DPH, #DMAORL A, INDX2	; set count back
1 1276 23	<u></u>		• • • • • • • • • • • • • • • • • • • •	RL	A.	s multiply by 2
2 1277 75 3 127A 84	FO	03		MOV	B, #3	; set up for divide by 3
127B C5	FO	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	DIV	AB A, B	<pre>5</pre>
5.127D B4	02 Fo	06		CUNE	A, #2, SF1	; IF remainder = 2 THEN
1280 05 7 1282 74	F0 00			INC	В	5 bump clock value
3 1284 80		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	MOV SJMP	A, #0 SF2	;
) 						, ELSE
) 1286 B4 1289 74	00	04	SF1:	CUNE	A, #0, SF2	; IF remainder = O THEN
1289 74 128B 15	02 F0		• • • • • • • • • • • • • • • • • • • •	MOV DEC	A, #2 B	; remainder := 2
128D C5	FO		SF2:	XCH	А, В	; decrement clock count ; switch back
128F F4				CPL	A	take 2's complement of (A)
1290 04				INC	A	
1291 02	AF			CLR	EA ⁻	; disable interrupts
				NOP	···· T// ·········	; give time to clear
1294 75 1297 C2	<u>A8</u>	82		MOV	IE, #IEDMA	; init for DMA
1297 C2 1299 C0	B4 82			CLR PUSH	DMATYPE	; set type for clear memory
TÎZ9B TÖÖ	83	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	PUSH PÜSH	DPL DPH	; Push DMA start address
: 129D 75	80	FF		MOV	THO, #OFFH	; onto the stack ; set timer O
12A0 F5	8A			MOV	TLO, A	; for number of characters
12A2 D2 12A4 22	8C	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	SETB RET	TRO	start timer running
1111				UE I		; start DMA

C:

C C

(

(((

<u>1</u>			••••		SERIAL	.m.senial.mont.inten	ruet_routine
3				;			
4				;	· · · · · · · · · · · · · · · · · · ·	FA . BUUGERRAGE. , & (). , 5,6().	ial sont interrupts.
5 6	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		ENTRY	 none	
7						•••••	
9				; ;	EXIT	none	
10 11				5	USES	none	
12	• • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	•••••	
13 12CB 14 12CD	ço	<u>D</u> O		SERIAL:	PUSH	PSW	; Save resisters
15.12CF	CO	FO			PUSH PUSH	ACC B	
16 12D1	có	82			PUSH	DPL	••••••
17.12D3. 18 12D5	Ç0 20	83 98	05	• • • • • • • • • • • • • • • • • • • •	P.USH	<u>PPH</u>	
19 12DS	20	99	38		JB JB	RI, SERIN TI, SEROUT	; Check receiver interrupt
20 12DB	80	ЗА			SJMP	SRT	5 Check transmit interrupt 7 This should never happen
21 22 12DD	75	39		SERIN:	MOV	AUTOCNT, #0	
2312E0.		24		***************************************	SETB	VSPF	; Change auto off
24 12E2 25 12E4	C2 E 5	98			CLR	RI	; Clear receiver int flas
26 12E6	54	99 7F	• • • • • • • • • • • • • • • • • • • •		MOV ANL	A, SBUF A, #01111111B	; Get character ; Mask off parity
2712E8.	29	43	92		JB	HAZ, SIO	, mask off parity ; IF not Hazeltine THEN
28 12EB	60	2A			JZ	SRT	; IF null THEN exit
30 12ED	20	23	03	SIO:	JB	PRNTF, SI1	; IF not printing THEN
31.12F0. 32.12F3	30 84	2E	24	SII:	JNB	ONLINE, SRT	<pre># IF offline THEN exit</pre>
33 12F6	20	23	06	511:	CJNE JB	A, #XOFF, SI2 PRNTF, SI1A	; IF -XOFF- received THEN
34 12F9	20	42	05	•••••	JB	ADM3, SI2	; IF not printing THEN ; IF ADM 3A THEN skip
35 12FC 36 12FF	20 D2	43	02		JB	HAZ, SI2	; IF Hazeltine THEN skip
37 1301	B4	11	02	SI1A: SI2:	SETB CUNE	XOFFROVED A, #XON, SI3	; flas for XOFF
38 1304	C2	16		×++	CLR	XOFFRCVED	; IF -XON- received THEN ; flas for no XOFF
391306 401309	20 12	23 01CA	<u>Ņ</u> Ē	SI3:	JB	PRNTF, SRT	; IF printing THEN exit
41 130C	30	D5	08		LCALL JNB	PCIF FO, SRT	; Put character in input FIFO
42 130F	71	C4			ACALL	DOXOFF	; IF -XOFF- needed THEN ; send an -XOFF- and exit
. 43 1311 44	80	04			SUMP	SRT	
45 1313	C2	99 13		SEROUT:	CLR	TI	: Clear transmit int floo
46 1315 47	D2	13			SETB	OKTRANS	; Clear transmit int flas ; Set ok to transmit
4/ 48 1317	DÖ	83		SRT:	POP	DPH	
49 1319	DO	82			POP	DPL	; Restore registers
50 131B	DO	FO			POP	В	••••••
.51 131D 52 131F	DO	E0		• • • • • • • • • • • • • • • • • • • •	POP	ACC PSW	
53 1321	71	26			ACALL	SRT1	; Double check
54 1323	D2	AF			SETB	EA	
55 1325 56 1326	22 C2	AF		SRT1:	RET		
57 1328	32	1 11		OKII.	CLR RETI	EA	; And return

729ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE 175

<u>.</u>					IRQ -	interrupt request	
2				; •	*TPO*	is evecuted when the C	RTC starts to display the 25th
j 1	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		line.	At this point the DMA	line number must have been set
5				;	to zer	o or some error has oc	cured. This routine is a safety
5	· · · · · · · · · · · · · · ·			;	check	for any such errors an	d will take at most one frome to
<u>,</u>				;	strais	hten out any such erro	rs. It also takes care of any
3 9)				; ;	clock	timine and blinkine.	
i					ENTRY	none	
2					EXIT		
3 						none	
5 					USES	none	
7 3 1329 -	co	Bo		IRQ:	PUSH	PSW	
3 1329 9 132B	CO			ING.	PUSH	ACC	; Save resisters
7 132D 7 132D	čŏ	<u>E0</u>			PUSH	В	•••••
1 132F		82			PUSH	DPL	
2 1331 3	co	82 83			PUSH	ĎРĤ	
4 1333	75	ö .			MOV	DMLNA, #0	; Safety check
5 6 1336	З1	84	• • • • • • • • • • • • • • • • • • • •		ACALL	CURSOR	; Display cursor
7 3 1338	90	2001			MOV	DPTR, #DSTAREG	
133B	ΕO				MOVX	A, @DPTR	; Clear any CRTC flags
)							
1330	05	38 37			INC	BLNKCNT	; Bump blink count
2 133E	E5				MOV	A, BLNKRATE	
3 1340 4 1342	60 B4	10	02		JZ CJNE	IRQ2 A, #OFFH, IRQ1	; IF not zero OR ; IF not full count THEN
+ 1342 5 1345	80 80		02		SJMP	IRQ2	; IF NOT FULL COUNT THEN
) 1343 5 1347	<u>85</u>	o <u>B</u>	80	IRQ1:	CJNE	A, BLNKCNT, IRQ2	; IF time to blink THEN
7 134A	75 	38	00	11/0/1	MOV	BLNKCNT, #0	reset count
134D	<u>í.ÿ</u> B2	01			CPL	BLINKF	togale blink flag
134F	12	1E18			CALL	WLATCH	
) 1352	05	3A		IRQ2:	INC	TCNT	; Bump time count
1354	E5	ЗА			MOV	A, TCNT	
1356	20	39 32	<u>05</u>		JB	FREQ, IRQ3	; IF 50 Hz AND 1 second OR
1359	B4	32	34		CUNE	A, #50, IRQ5	A TE /O US OND 1 SECOND TURN
5.135C 5.135E	80 B4	03 3C	2F	IRQ3:	SJMP CJNE	IRQ4 A, #60, IRQ5	; IF 60 Hz AND 1 second THEN
7 1361		30 3A	00	IRQ3: IRQ4:	MOV	TCNT, #0	: initialize count
1364	75 	≳∺ 3B	XY	AD%T*	INC	TSEC	; initialize count ; bump seconds
1366	E5				MOV	A, TSEC	, pame seconds
1368	E5 B4	3B 3C	25		CUNE	A, #60, IRQ5	; IF one minute THEN
136B	30	3B	OB		JNB	SCRNSAVE, IRQ4A	; IF screen saver THEN
2 136E	05	39			INC	AUTOCNT	; IF screen saver THEN ; bump auto off
3 1370	E5	39			MOV	A, AUTOCNT	
1372	В4	OF	04		CJNE	A, #15, IRQ4A	; IF saver time up THEN
5.1375	ç2	24			ÇLR	YSPF	flas it is off
5 1377	D2	92			SETB	VSP	; and turn off display

ZZ9ROM; MICRO	BENCH 80	51 CROSS	ASSEMBLER	(V1)-218 24-	FEB-83 15	5:30:06 PAGE 176	• • • • • • • • • • • • • • • • • • • •		
*** Z-29 COMPI 1 1379 2 1370	VIER TER 75 	MINAL.EI	RMWARE.YERS	ION.1,03.***.	MOV INC	•••••		inițialize seconds. bump minutes	
3 137E 4 1380	E5 B4	30	OD		MOV	A, TMIN A, #60, IRQ5	; 	IF one hour THEN	
5 1383 6 1386	75 05	3C	00		MQV	IMIN;.#Q THOUR	;	initialize minute bump hours	5
7 1388 8 138A 9 138D	E5 B4 75	3D 18 3D	03 00		MOV	A, THOUR A, #24, IRQ5	· · · · · · · · · · · · · · · · · · ·	IF 24 hours THEN	
9 138D 10 11 1390	í Do			IRQ5:	MOY	DPH #0		initialize_hour estore_resisters_and_r	5
12 1392 13 1394	DO DO	83 82 F0 E0			POP POP	DPL B		esrace Weststers Bud U	ár n. u
14 1396 15 1398 16 139A	DO DO 32	EO DO			POP POP	ACC PSW			
10 137н					RETI				
									•••••
•••••	• • • • • • • • • • • • • • • • • • • •						•••••		
•••••						,	• • • • • • • • • • • • • • • • • • • •		
••••••	• • • • • • • • • • • • •		•••••			•••••			•••••
***************************************	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •				
							••••••		•••
•••••	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •					
			•••••						
			• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •				
•••••		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •					•••••	
•••••		• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •				
		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •				
			• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		
***************************************									••••••
									••••

1					TRANSM	IT — transmit to host					
2 3				;	*TRANSMIT* waits for an empty output holding register, sets						
í 1				*	Parity	, and outputs the char	acter to the serial port.				
				; ;	ENTRY	(A) = character to o	uteut				
				; ;	EXIT	(A) = character with	correct parity				
· · · · · · · · · · · · · · · · · · ·				; 	USES	Α					
2 3 1 139 6	20	1%	FD	·····transmit:····	JB	···XOFFRCVED; TRANSMIT	; Wait until no -XOFF-				
5											
139E			-00	TRANS:	CLR	-	; Start with no parity ; IF parity enabled THEN				
139F	30	28	OA		JNB SETB	PRTYENA, TRA	; do set parity				
13A2	20	20	06		JB	PRTYSTK, TRA	; IF not stick parity THEN				
13A3 13A6	20 A2	29 bo	99		MÖV	C, P	do even parity				
13A8		2A	01		JB	PRTYEVN, TRA	; IF not even parity THEN				
13AB	20 B3				CPL	c	; do odd parity				
13AC	92	E7		TRA:	MOV	ACC.7, C	; Put in parity bit				
13AE	20	13 99	63 FA	TRW:	JB	OKTRANS, TROK	; IF ok to transmit OR				
13B1	30	99	FA		JNB	TI, TRW	; IF transmit buffer empty THEN				
13B4 13B7	<u>20</u>	02 3É	03	TROK:	JB	PORTF, TROK1	; IF auxiliary port OR ; IF hardware handshake THEN				
: 13B7	30	3E	03	77 F	JNB	HNDSHK, TROK2					
13BA	20	B5	F1	TROK1:	JB	CTS, TRW	; IF not CTS THEN do asain				
) 13BD	CO	00		TROK2:	CLR	TI	; clear buffer empty flas				
13BF	C2 C2	99 13			CLR	OKTRANS	; make not ok to transmit				
1301	F5	99			MOV	SBUF, A	; output data				
1303	F5 22				RET						
											

)))))))

					POXOFF	dosend.XQFFto.ho	st
					D0XQF	F.ereemets.outgoing.	data and outputs an XOFF to host
5				;	settin	a the proper flags.	
6 7				, , , , , , , , , , , , , , , , , , ,	ENTRY		
8				·······			••••••
9 10	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	····	EXIT	none	
<u>11</u>	• • • • • • • • • • • • • • • • • • • •				UŞEŞ	A	
13 14 1304	<u>20</u>	10	1A	DOXOFF:	JB		··········· <u>···</u> ······················
15 1307		10			JB SETB	XOFFSENT, SXR XOFFSENT	; IF XOFF already sent THEN exit; Flag that handshake was done
16 17 1309	30 	3E	. 05		JNB	HNDSHK, XMTXOFF	; IF hardware handsbake THEN
18 13CC		95 94			SETB	RTS	; set handshake lines
19 13CE 20 13DO	<u>D2</u> 22		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	SETB RET	DTR	•••••••••••••••••••••••••••••••••••••••
.21 22 13D1	74	13		XMTXOFF:	MOV	A, #XOFF	; ELSE output an XOFF
23 13D3 24	61	9E		• • • • • • • • • • • • • • • • • • • •	AJMP	TRAN\$	
25 26	• • • • • • • • • • •						
27 28							
29				; ;		- do send XON to host	
30 31				; •	*DOXON	* preempts outsoins d	ata and outputs an XON to host
32		• • • • • • • • • • • • • • • • • • • •		;	cüácki	ngtnatlt.doesnot.o	yerride any user XOFF.
33 34		• • • • • • • • • • • • • • • • • • • •			ENTRY	none	
35 36		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	5	EXIT	none	
37 38	• • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		USES	A	•••••
39 40						н	
41 13D5 42 13D8	30 	10	09	DOXON:	JŅB	XOFFSENT, SXR	; IF no XOFF sent THEN exit
42 13D8 43		10			CLR	XOFFSENT	; IF no XOFF sent THEN exit ; Clear flas on handshake
43 44 13DA 45 13DD	30	3E	05		JNB	HNDSHK, SXON1	; IF hardware handshake THEN
46 13DF	C2 	95 94		• • • • • • • • • • • • • • • • • • • •	CLR CLR	RTS DTR	; clear handshake lines
47 13E1 48	22	• • • • • • • • • • • • • • • • • • • •		SXR:	RET		
49 13E2 50 13E5	20 74	11	FC	SXON1:	JB MOV	UXOFFSENT, SXR	; IF no user XOFF THEN
51 13E7	61	9E		XMTXON:	MOV AJMP	A, #XON TRANS	<pre>; send an XON ; transmit it and return</pre>
	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,						
				••••••			

				LOOKUP	— table lookup and	jump
2 .3			; ;			try in the table passed to it. If
4 5			;			is removed and control siven to the en the main table is searched. If
		• • • • • • • • • • • • • • • • • • • •				is removed and control given to the
. 7			\$	routine	e. If not found th	en control is siven back to the
8 9 10				callins	o routine.	
10 11 12			; ;	ENTRY	(B) = table lenst	
12 13 14			\$ \$ 		(DPTR) = beginnin	
14 15 16			; ;	EXIT	return if not for	nd in tables
17			; 	USES	ail	
18 19 13E9	91 50	04 0B	LOOKUP:	ACALL	XSTABX	; Search users table
20 13EB 21 13ED	50 75	0B F0 13	2	JNC MOV	LKUP1 B, #LOTL	; IF not found THEN
21 13ED 22 13FO	90	1432		MOV	DPTR, #LOT	
23 13F3	91 50	04 01		ACALL JNC	XSTABX	; search main table ; IF not found THEN
24 13F5 25 13F7	22			RET	LKUP1	; return
26 27 13F8	DO DO	EO	LKUP1:	POP	ACC	; ELSE found THEN
28 13FA 29 13FC	DO E4	EO		POP CLR	ACC A	; remove return address
22 13FC 30 13FD	53		,	JMP	@A+DPTR	; jump to routine

))))))))))

***.Z-29.CQM						
1				XSTXtransmit.str	ing to line 25 following CALL	• • • • • • • • • • • • • • • • • • • •
3 4 5			7 7 7	bytes so that routi	o the routine *XST*. It is used to save thes may use a short jump to here instea the other routine.	d
7 8 13FE 9		1B9D	XSTX:	LJMP XST	; Jump to routine	
10 11						
12 13 14			5.5	XSTATXtransmit.s	string to line 25	
15 16 17 18				save bytes so that	to the routine *XSTAT*. It is used to routines may use a short jump to here ump to the other routine.	
19 20 1401 21	02	1BA8	XSTATX:	LJMP XSTAT	; Jump to routine	
22 23 2 4						
25 26 27			7	XSTABX.m.search.tab		
27						
28 29			,	save bytes so that	to the routine *STAB*. It is used to routines may use a short jump to here ymp to the other routine.	
28 29 30 31				save bytes so thatinstead of a long j	routines may use a short jump to here ump to the other routine.	
28 29 30			,	save bytes so that	routines may use a short jump to here	
28 29 30 31 32 1404				save bytes so thatinstead of a long j	routines may use a short jump to here ump to the other routine.	
28 29 30 31 32 1404				save bytes so thatinstead of a long j	routines may use a short jump to here ump to the other routine.	
28 29 30 31 32 1404				save bytes so thatinstead of a long j	routines may use a short jump to here ump to the other routine.	
28 29 30 31 32 1404				save bytes so thatinstead of a long j	routines may use a short jump to here ump to the other routine.	
28 29 30 31 32 1404				save bytes so thatinstead of a long j	routines may use a short jump to here ump to the other routine.	
28 29 30 31 32 1404				save bytes so thatinstead of a long j	routines may use a short jump to here ump to the other routine.	
28 29 30 31 32 1404				save bytes so thatinstead of a long j	routines may use a short jump to here ump to the other routine.	
28 29 30 31 32 1404				save bytes so thatinstead of a long j	routines may use a short jump to here ump to the other routine. ; Jump to routine	
28 29 30 31 32 1404 33	02	013A	XSTABX:	save bytes so that instead of a long j LJMP STAB	routines may use a short jump to here ump to the other routine. ; Jump to routine	
28 29 30 31 32 1404 33	02	013A	XSTABX:	save bytes so that instead of a long j LUMP STAB	routines may use a short jump to here ump to the other routine. ; Jump to routine	
28 29 30 31 32 1404 33	02	013A	XSTABX:	save bytes so that instead of a long j LJMP STAB	routines may use a short jump to here ump to the other routine. ; Jump to routine	

ŻŹŚRÓM; MICRÓB *** Z−29 COMPU	ENCH 805	1 CROSS ASSE IINAL FIRMWAR	MBLER (V1)-218 24 E VERSION 1.03 ***	-FEB-83 15:	:30:06 PAGE 181		
1				SFCKB -	- setup fetch character	from keyboard	
2 3 4 5				*SFCKB: not re and the	* waits for characters turn until a character e characters that follo	from the keyboard. It does is returned. All access codes we are dropped.	
6			· · · · · · · · · · · · · · · · · · ·				
7 8			· · · · · · · · · · · · · · · · · · ·	ENTRY	none		
10				EXIT	(A) = character from	keyboard	
				USES	A, B, DPTR		
12 13 14						# Unit Con Low	
15 1407 16 140A	<u>12</u>	02B6 FB	SFCKB:	CALL	FCKB SFCKB	; Wait for key	
17 140C 18 140F	B4 12	9F 07 02B6	SFCKB1:	CUNE CALL	A, #KB_ACC, SFCKB2 FCKB	; IF access code THEN ; drop next character	
19 1412	40	FB Fi		JC SJMP	SFCKB1 SFCKB	; and keep on soins	
20 1414 21 1416	80 22	F1	SFCKB2:	RET		; ELSE return	
				············			
					,		

)

)

•

)

)

1				SETUP1	setue mode line.	#1
2 3			,			l setup mode program.
4 5			;		O OVE P P P VE. 178.	C. SEVAL MARE CLASSENIA
6 7			,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	ENTRY	none	
8			**************************************	EXIT	none	
10 11		•••••••••••••••••••••••••••••••••••••••		USES	all	
12					• • • • • • • • • • • • • • • • • • • •	
13 1417 14 141A 15	20 22	9301	SETUP:	J₽ RET	SETLCK, SETUPO	i Check for locked setup
16 141B	12	1B2C	SETUPO:	CALL	SCLRLR	; Setup clear line reverse
18 141E	91	<u>6</u> E	SETUP1:	ACALL	L1	; Display first line
19.1420. 20.1422	91 75	Q7	STP1A:	AÇAĻĻ. MOV	SFCKB B, #L1TL	
211425.	90	146B		MQY	DPTR:#L1T	
22 1428 23.142A.	71 80	E9 F4		ACALL	LOOKUP	; Lookup command
24 25			••••••		STP.1A	i Keep on going
49 26 27	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	.	Tossle	on/off line	
28 1 42 C	B2	2E	STP1B:	CPL	ONLINE	; Toggle on/off line
29.142E. 30.1430	31 80	9E EC		CALL SJMP	INITKB SETUP1	; Reconfigur keyboard
31 32 33						
3 4 35				LOT	every line table	
36 37			; ;		contains the jump to	able for every line.
38 39 40	1432		LOT		\$	
41 1432	OD 141E			DB	CR	
42 1433				DB DW	CR SETUP1	; Main setup line
43 44 1435	61	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	DB	······································	
45 .1436 46	141E		• • • • • • • • • • • • • • • • • • • •	DW	SETUP1	; Main setup line
47 1438	41 141E			DB DW	/A/ SETUP1	
48 1439 49						; Main setup line
49 50 143B 51 143C	74	• • • • • • • • • • • • • • • • • • • •	••••••	DB	····> t -····································	•••••••••••••••••••••••••••••••••••••••
52	1407	• • • • • • • • • • • • • • • • • • • •		DW	SETUP2	; Set tabs
53 143E 54 143F	54 14D7			DB	/T/ SETUP2	
55				DW		; Set tabs
56 1441 57 1442	62	• • • • • • • • • • • • • • • • • • • •	•••••	DB		
971444	1542			DW	SETUP3	; Protocol #1

((

(

1444	42		DB	<u>B</u>	
1445	1542		DW	SETUP3	; Protocol #1
1447 1448	63 1680		DB DW	´c´ SETÜP4	; Protocol #2
144A	43		DB		
1448	16B0		DW	SETUP4	; Protocol #2
144D 144E	64 1795		DB DW		7 Misc #1
1450	44		DB	νDν	
1451	1795		DW	SETUP5	; Misc #1
1453 1454	65 1865		DB DW		; Keyboard
1454			DB DW		; Kerbodra
1456 1457	45 1865		DB DW	'E' SETUP6	; Keyboard
! ! . 1.4 59	<u> </u>		DB	fr	
145A	18F4		DW	SETUP7	; Misc #2
145C 145D	46 18F4		DB DW	FF SETUP7	7 Misc #2
}					
145F 1460	19DD		DM DB	SETUP8	; Misc #3
1462	47		DB DB	/G/ SETUP8	; Misc #3
1.1443	1900				
1445 1466	88 1AF6		DB DW	KB_SETUP SETUPRET	; Setue key ; Return
7 3 1468	cs		DB	KB_SETUP + KB_SHFT	; Shift setup key
? 14 69	1,802		W.Q	SETUPSAYE	; Saye and return
 !			FQU	(\$-LQT)/3	; Table #0 length
			L1T -	line #1 table	
			L1T	contains the jump table	e for line #1 in setup mode.
'		L1T	EQU	 \$	
146B	31		DB		; Option #1
. 1460 3.1460	1420		DW	STP1B	i Tosale on/off line
* 5	0001	L1TL	EQU	(集-L1T)/3	: Table length

_								
<u>1</u>								• • • • • • • • • • •
3						ine #1		
5				, 	*L1* d	iselays setue informa	ation for line #1.	
6 7				; ;				
8				*	ENTRY	none		••••••
10					EXIT	none		• • • • • • • • • • • • • • • • • • • •
11					ÜŚĔŚ	all		
13				·····				
14 15 146E	E4			L1:	CLR	Α	; Position	
16				• • • • • • • • • • • • • • • • • • • •				
17 146F 18 1471	71 2A	FE 2A	20		ACALL.	XSTX *** SETUP MENU A **	* (Ver 1.03) 1.7, (7.7+80H)	
1474 1477	53 55	45 50	54 20					
147A	4D 55	45 20	4E			•••••		
147D 1480	55 20	20 24	41					
1483	20 20	2A 20	28			· · · · · · · · · · · · · · · · · · ·		
1486	56 20	65 31	2A 28 72 2E	• • • • • • • • • • • • • • • • • • • •				• • • • • • • • • • • • • • • • • • • •
148C	30 20	33 20	29 20					
148F 1492 19	31	20 2E	20 A0					
19					MOV	DPTR, #LN1B1		
20 1495 21 1498	90 20	14C7 2E	93		JB	ONLINE, L1A	; Get proper message for	• • • • • • • • • • • • • • • • • • • •
.22 149B .23 149E	90 91	14CF.		Lia:	MOV ACALL	DPTR, #LN1B2 XSTATX	; on/off line ; Next part	
24 25 14A0	71	FE						
26 14A2	20 20	20	20 45		ACALL DB	XSTX MENUS -A- to -	-G- or -T- for TAB', ('S'+80H)	
14A5 14A8	20 4E	4D	45	•••••				
1448	<u>4E</u>	55 2b	53 		• • • • • • • • • • • • • • • • • • • •	•••••		• • • • • • • • • • • • • • • • • • • •
14AE 14B1	2D 6F	<u>20</u>	74 2D					
14B4	47	2D	20					
14B7 14BA	6F	・・・・フ2・・・ 54	20 2D			,		
1480	2D 20	99	2D					• • • • • • • • • • • • • • • • • • • •
14C0 14C3	72 41	20 42	54 b3					• • • • • • • • • • • • • • • • • • • •
27					<u></u>			
28 1406 29	22				ŔĔŤ			
∵36`····· 31	• · · · · · · · · · · ·			•	Line #	l setup mode messages		
"32"14C7"	6F	6 <u>E</u>	₂₀	LNTBT:	BQ	on line', (2.7+80)	4)	
14CA 14CD	6C 65	69 Ao	6E					
33 14CF	6F	66	66	LN1B2:	DB	'off lin', ('e'+80H	1)	

C

C

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE 184+ *** Z-29 COMPUTER TERMINAL FIRMWARE VERSION 1.03 ***
14D2 20 6C 69 14D5 6E E5

)

)

)

)

•

)

1					; ;	SETUP2	setup mode line #2	
2					;			••••••
3 4	• • • • • • •			• • • • • • • • • • • •		*SETUF	2*.is.the.setue.mode.lin	e for set/clear tabs.
. 5					,	-F1R2-	· is used as the cursor p	osition.
6								
<u>7</u>						ENTRY	none	
9					;	EXIT	5050	
? 10					· · · · · · · · · · · · · · · · · · ·		none	***************************************
.11						UŞEŞ	all	
12 13								
14 14	1D7	79			SETUP2:	MOV	PTR2, #0	; -PTR2- is cursor position
15 14	ID9	B1	OD 		STP2A:	ACALL	L2	; Display line #2
16 14		91			STP2B:	CALL	SFCKB	; Fetch character from keyboard
17 14 18 14		75 90	F0 1501	04		MOV	B, #L2TL	
18 14 19 14		7.1				MOV ACALL	DPTR, #L2T	• Laste was a second
20 14	15.7 165	 80	E9	• • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	SJMP	LOOKUP STP2B	; Look up command ; Keep on going
21								/ ISSUE ON DOLLID
22								
.23 24			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		Move 1	eft	
25 14	E7	19			STP2C:	DEC	PTR2	; Decrement
26 14	E8	B9	FF	01		CUNE	PTR2, #(0-1), STP2C1	; IF wrap around THEN
27 14	EB	09 80				INC	PTR2	; bump back up
28 14	FEC	80	EB		STP2C1:	SJMP	STP2A	; keep going
29 30			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •				
31					;	Move r	isht	
32			• • • • • • • • • • • • • • • • • • • •		;			
33 14 34 14		<u>69</u>	50	61	STP2D:	INC	PTR2	; Bump
35 14		19	50	01		DEC DEC	PTR2, #80, STP2D1 PTR2	; IF wrap around THEN
36 14	F3		È4	• • • • • • • • • • • • • • • • • • • •	···STP2D1:	SJMP	STP2A	; bump back down ; keep going
37 38							3.1. 2.1.	, Keer soins
								•••••••••••••••••••••••••••••••••••••••
39 40						Set ta	b	
41 14	F5	D2	D5		; STP2E:	SETR	FO	
42 14	F7····	D2 80	05 02		No. 1 1 America *	SETB SJMP	F0 STP2F1	; Set tab
43 44							_	
44 45						63	A b.	
45 46			• • • • • • • • • • • • • • • • • • • •			Clear	tab	
47 14	F9	C2	D5		, STP2F:	CLR	FO	
48 14		<u>E</u> ğ			STP2F1:	MGV	FO A, PTR2	; Get Position
19 14		12	05A2			CALL	SETTAB	; Set/clear tab
50 14	FF	80	DS			SJMP	STP2A	; Keep going
51 52								
53								
54					•••••	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	••••••
55 56					5.5	L2T -	line #2 table	
					,			
57						*L2T*	contains the jump table	for line #2 in setup mode.

ENTRY none							
1501 82		1501		L2T	EQU	\$	
1502	} (***\$84******	ga			ne	KB LEET	: Cursor left
1504 83							
1507 80)						
1507 80	7 1504	83			DB	KB_RIGHT ····≈+b>h	; Cursor right
1507 80		1466					, 110ve 1 1511t
150A SI							
150A 81 DB KB_DDWN Cursor down 150B 14F9 DW ST2F 1 Clear tab 1 C	1508	14F5			DW	STP2E	; Set tab
1		81			DB	KB_DOWN	; Cursor down
	150B				DW		; Clear tab
	j.						
		0004		LZTL	FØU	(\$-L21)/3	; lable length
	}						
	, 						
)				12 - 1	line #2	
		• • • • • • • • • • • • • • • • • • • •					
ENTRY none	l				*L2* c	displays setup informa	tion for line #2 set/clear tabs.
EXIT				;			
EXIT							
150D 90 5780 L2: MOV DPTR #L25MEM Work with 25th line 1510 70 00 MOV TEMP, #0 Counter variable 1512 78 29 MOV PTR1 #TABTAB Point to tab table 1514 7A 0A MOV INDX1 #10 10 bytes 1515 7B 08 MOV A EPTR1 REPEAT get byte 1517 7B 08 MOV INDX2 #8 8 bits 1519 FE MOV MORK1 A saye value for later 1518 04 INC A fere and adjust pos 1510 75 FO OA MOV B #10 calculate characte 15120 E5 FO MOV A B E520 E5 FO MOV A B E520 E5 FO MOV A B E525 ED MOV A TEMP TEMP TEMP) 				ENTRY	none	
150D 90 5780 L2: MOV DPTR, #L25MEM ; Work with 25th line 1510 7D 00 MOV TEMP, #0 ; Counter variable 1512 78 29 MOV PTR1, #TABTAB ; Point to tab table 1514 7A 0A MOV INDX1, #10 ; 10 bytes 1516 E6 L2A: MOV A, @PTR1 ; REPEAT get byte 1517 7B 08 MOV INDX2, #8 ; 8 bits 1519 FE MOV WORK1, A ; save value for later 1518 ED L2B: MOV A TEMP ; REPEAT 1518 O4 INC A ; set and adjust Pos 1510 75 FO 0A MOV B, #10 ; calculate characte 1517 34 DIV AB 1520 E5 FO MOV A, B 1522 24 30 ADD A, #/O/ ; convert to ASCII 1524 FF MOV WORK2, A ; and save it 1525 ED MOV A, TEMP 1526 B5 01 02 CUNE A, PTR2A, L2C ; IF cursor = Positi 1529 7F 09 MOV WORK2, #09H ; change to cursor				***************************************			
150D 90 5780 L2: MOV DPTR, #L25MEM ; Work with 25th line 1510 7D 00 MOV TEMP, #0 ; Counter variable 1512 78 29 MOV PTR1, #TABTAB ; Point to tab table 1514 7A 0A MOV INDX1, #10 ; 10 bytes 1516 E6 L2A: MOV A, @PTR1 ; REPEAT get byte 1517 7B 08 MOV INDX2, #8 ; 8 bits 1519 FE MOV WORK1, A ; save value for later 1518 ED L2B: MOV A TEMP ; REPEAT 1518 O4 INC A ; set and adjust Pos 1510 75 FO 0A MOV B, #10 ; calculate characte 1517 34 DIV AB 1520 E5 FO MOV A, B 1522 24 30 ADD A, #/O/ ; convert to ASCII 1524 FF MOV WORK2, A ; and save it 1525 ED MOV A, TEMP 1526 B5 01 02 CUNE A, PTR2A, L2C ; IF cursor = Positi 1529 7F 09 MOV WORK2, #09H ; change to cursor	· 			7 7			
150D 90 5780 L2: MOV DPTR, #L25MEM ; Work with 25th line 1510 70 00 MOV TEMP, #0 ; Counter variable 1514 74 04 MOV INDX1, #10 ; 10 bytes 1516 E6 L2A: MOV A, @PTR1 ; REPEAT get byte 1517 78 08 MOV INDX2, #8 ; 8 bits MOV MORK1, A ; save value for later 1518 ED L2B: MOV A, TEMP ; REPEAT REPEAT REPEAT REPEAT 1518 O4 INDX B, #10 ; calculate characte 1510 75 F0 OA MOV B, #10 ; calculate characte 1520 E5 F0 MOV A, B 1522 24 30 ADD A, #70′ ; convert to ASCII 1524 FF MOV A, TEMP ; and save it MOV A, TEMP ; calculate characte 1526 B5 O1 O2 CUNE A, PTR2A, L2C ; IF cursor = Positi D1529 7F O9 MOV A, WORK1 ; restore tab table 1528 EE L2C: MOV A, WORK1 ; restore tab table 1528 EE L2C: MOV A, WORK1 ; restore tab table 1528 EE L2C: MOV A, WORK1 ; restore tab table 1528 EE L2C: MOV A, WORK1 ; restore tab table 1528 EE L2C: MOV A, WORK1 ; restore tab table 1528 EE L2C: MOV A, WORK1 ; restore tab table 1528 EE L2C: MOV A, WORK1 ; restore tab table 1528 EE L2C: MOV A, WORK1 ; restore tab table 1528 EE L2C: MOV A, WORK1 ; restore tab table 1528 EE L2C: MOV A, WORK1 ; restore tab table 1528 EE L2C: MOV A, WORK1 ; restore tab table 1528 EE L2C: MOV A, WORK1 ; restore tab table 1528 EE L2C: MOV A, WORK1 ; restore tab table L2C: MOV A, WORK1 ; restore table L2C: L2C: L2C: L2C: L2C: L2C: L2C: L2C	, 				EXIT	none	
1510 70	, , ,			; ;	EXIT	none	
1512					EXIT	none all	* Work with 25th line
1516 E6		90	5780 00	.; ; ; ; ;	EXIT USES MOV	none all DPTR, #L25MEM	
1516 E6	150D	90 70		.; ; ; ; ; L2:	EXIT USES MOV MOV	none all DPTR, #L25MEM TEMP, #O	; Counter variable
1517 7B 08	: 150D 1510 : 1512 : 1514	90 70 78 78			EXIT USES MOV MOV MOV	none all DPTR, #L25MEM TEMP, #O PTR1, #TABTAB INDX1, #10	; Counter variable ; Point to tab table ; 10 bytes
151B 04 1NC A	: 150D 1510 1512 1514 1516	78 7A E6	29 0A		EXIT USES MOV MOV MOV MOV MOV	none all DPTR, #L25MEM TEMP, #O PTR1, #TABTAB INDX1, #10 A, @PTR1	; Counter variable ; Point to tab table ; 10 bytes ; REPEAT get byte
151B 04 1NC A 5 9et and adjust Pos 151C 75 FO OA MOV B, #10 5 calculate characte 151F 84 DIV AB	150D 1510 1512 1514 1516 1517	78 7A E6 7B	29 0A		EXIT USES MOV MOV MOV MOV MOV MOV	none all DPTR, #L25MEM TEMP, #0 PTR1, #TABTAB INDX1, #10 A, @PTR1 INDX2, #8	; Counter variable ; Point to tab table ; 10 bytes ; REPEAT get byte ; 8 bits
151F 84	150D 1510 1512 1514 1516 1517	78 7A E6 7B	29 0A	L2A:	EXIT USES MOV MOV MOV MOV MOV MOV MOV MOV	none all DPTR, #L25MEM TEMP, #0 PTR1, #TABTAB INDX1, #10 A, @PTR1 INDX2, #8 WORK1, A	; Counter variable ; Point to tab table ; 10 bytes ; REPEAT get byte ; 8 bits ; save value for later
151F 84	150D 1510 1512 1514 1516 1517 1519	78 7A E6 7B FE ED	29 0A	L2A:	EXIT USES MOV MOV MOV MOV MOV MOV MOV MOV MOV MO	none all DPTR, #L25MEM TEMP, #O PTR1, #TABTAB INDX1, #10 A, @PTR1 INDX2, #8 WORK1, A A, TEMP	; Counter variable ; Point to tab table ; 10 bytes ; REPEAT set byte ; 8 bits ; saye value for later ; REPEAT
151F 84	150D 1510 1512 1514 1516 1517 1519	78 7A E6 7B FE ED	29 0A 08	L2A: L2B:	EXIT USES MOV MOV MOV MOV MOV MOV MOV MO	none all DPTR, #L25MEM TEMP, #O PTR1, #TABTAB INDX1, #10 A, @PTR1 INDX2, #8 WORK1, A A, TEMP A	; Counter variable ; Point to tab table ; 10 bytes ; REPEAT get byte ; 8 bits ; save value for later ; REPEAT ; get and adjust position
1522	150D 1510 1512 1514 1514 1517 1517 1519 1518	78 7A E6 7B FE ED 04	29 0A 08	L2A: L2B:	EXIT USES MOV MOV MOV MOV MOV MOV MOV MO	none all DPTR, #L25MEM TEMP, #0 PTR1, #TABTAB INDX1, #10 A, @PTR1 INDX2, #8 WORK1, A A, TEMP A B, #10	; Counter variable ; Point to tab table ; 10 bytes ; REPEAT set byte ; 8 bits ; saye value for later ; REPEAT
1522	150D 1510 1512 1514 1514 1517 1517 1519 1518	78 7A E6 7B FE ED 04	29 0A 08 F0 06	L2A: L2B:	EXIT USES MOV MOV MOV MOV MOV MOV MOV MO	none all DPTR, #L25MEM TEMP, #0 PTR1, #TABTAB INDX1, #10 A, @PTR1 INDX2, #8 WORK1, A A, TEMP A B, #10 AB	; Counter variable ; Point to tab table ; 10 bytes ; REPEAT get byte ; 8 bits ; save value for later ; REPEAT ; get and adjust position
, 3 1525 ED MOV A, TEMP 9 1526 B5 01 02 CUNE A, PTR2A, L2C ; IF cursor = Positi 3 1529 7F 09 MOV WORK2, #09H ; change to cursor 2 152B EE L2C: MOV A, WORK1 ; restore tab table	150D 1510 1512 1514 1514 1516 1517 1519 1518 1518 1518 1516 151F	78 7A E6 7B FE ED 04 75 84 E5	29 0A 08 F0 06	L2A: L2B:	EXIT USES MOV MOV MOV MOV MOV MOV MOV MO	none all DPTR, #L25MEM TEMP, #O PTR1, #TABTAB INDX1, #10 A, @PTR1 INDX2, #8 WORK1, A A, TEMP A B, #10 AB A, B	; Counter variable ; Point to tab table ; 10 bytes ; REPEAT get byte ; 8 bits ; save value for later ; REPEAT ; get and adjust position ; calculate character
1525 ED	150D 1510 1512 1514 1516 1517 1519 1518 151B 151C 151F 1520	78 7A E6 7B FE ED 04 75 84 E5	29 0A 08 F0 06	L2A: L2B:	EXIT USES MOV MOV MOV MOV MOV MOV MOV INC MOV DIV ADD	none all DPTR, #L25MEM TEMP, #O PTR1, #TABTAB INDX1, #10 A, @PTR1 INDX2, #8 WORK1, A A, TEMP A B, #10 AB A, B	; Counter variable ; Point to tab table ; 10 bytes ; REPEAT set byte ; 8 bits ; save value for later ; REPEAT ; set and adjust position ; calculate character ; convert to ASCII
2 1526 B5 01 02 CJNE A, PTR2A, L2C ; IF cursor = Positi 5 1529 7F 09 MOV WORK2, #09H ; change to cursor 2 152B EE L2C: MOV A, WORK1 ; restore tab table	150D 1510 1514 1514 1514 1516 3 1517 9 1518 1518 2 151C 3 151F 1 1520 5 1522	78 7A E6 7B FE ED 04 75 84 E5	29 0A 08 F0 06	L2A: L2B:	EXIT USES MOV MOV MOV MOV MOV MOV MOV INC MOV DIV ADD	none all DPTR, #L25MEM TEMP, #O PTR1, #TABTAB INDX1, #10 A, @PTR1 INDX2, #8 WORK1, A A, TEMP A B, #10 AB A, B	; Counter variable ; Point to tab table ; 10 bytes ; REPEAT set byte ; 8 bits ; save value for later ; REPEAT ; set and adjust position ; calculate character ; convert to ASCII
1529 7F 09 MOV WORK2,#09H ; change to cursor 152B EE L2C: MOV A, WORK1 ; restore tab table	150D 1510 1512 1514 1516 1517 1519 1518 1518 1510 1515 1520 1522	78 7A E6 7B FE ED 04 75 84 E5 24	29 0A 08 F0 06	L2A: L2B:	EXIT USES MOV MOV MOV MOV MOV MOV INC MOV DIV MOV ADD MOV	none all DPTR, #L25MEM TEMP, #0 PTR1, #TABTAB INDX1, #10 A, @PTR1 INDX2, #8 WORK1, A A, TEMP A B, #10 AB A, B A, B A, #<0/	; Counter variable ; Point to tab table ; 10 bytes ; REPEAT set byte ; 8 bits ; save value for later ; REPEAT ; set and adjust position ; calculate character ; convert to ASCII
: 152B EE L2C: MOV A, WORK1 ; restore tab table	150D 1510 1512 1514 1514 1516 1517 1519 1518 151B 151C 151F 1520 1522 1524	78 7A E6 7B FE ED 04 75 84 E5 24 FF	29 0A 08 F0 06 F0 30	L2A: L2B:	EXIT USES MOV MOV MOV MOV MOV INC MOV DIV MOV ADD MOV	none all DPTR, #L25MEM TEMP, #0 PTR1, #TABTAB INDX1, #10 A, @PTR1 INDX2, #8 WORK1, A A, TEMP A B, #10 AB A, B A, B A, #'0' WORK2, A A, TEMP	; Counter variable ; Point to tab table ; 10 bytes ; REPEAT set byte ; 8 bits ; save value for later ; REPEAT ; set and adjust position ; calculate character ; convert to ASCII ; and save it
152B EE L2C: MOV A, WORK1 ; restore tab table	150D 1510 1512 1514 1514 1516 1517 1519 1518 1516 1516 1516 1520 1522 1524	78 7A E6 7B FE ED 04 75 84 E5 24 FF	29 0A 08 F0 06 F0 30	L2A: L2B:	EXIT USES MOV MOV MOV MOV MOV MOV INC MOV DIV MOV ADD MOV CJNE	none all DPTR, #L25MEM TEMP, #0 PTR1, #TABTAB INDX1, #10 A, @PTR1 INDX2, #8 WORK1, A A, TEMP A B, #10 AB A, B A, #'O' WORK2, A A, TEMP A, PTR2A, L2C	; Counter variable ; Point to tab table ; 10 bytes ; REPEAT get byte ; 8 bits ; save value for later ; REPEAT ; get and adjust position ; calculate character ; convert to ASCII ; and save it
01.0 A	150D 1510 1512 1514 1516 1517 1518 1518 151C 151C 151F 1520 1522 1524	78 7A E6 7B FE ED 04 75 84 E5 24 FF ED B5 7F	29 0A 08 F0 06 F0 30	L2A: L2B:	EXIT USES MOV MOV MOV MOV MOV MOV INC MOV DIV MOV ADD MOV CJNE	none all DPTR, #L25MEM TEMP, #0 PTR1, #TABTAB INDX1, #10 A, @PTR1 INDX2, #8 WORK1, A A, TEMP A B, #10 AB A, B A, #'O' WORK2, A A, TEMP A, PTR2A, L2C	; Counter variable ; Point to tab table ; 10 bytes ; REPEAT get byte ; 8 bits ; save value for later ; REPEAT ; get and adjust position ; calculate character ; convert to ASCII ; and save it
1 1520 33 KUL A	150D 1510 1512 1514 1516 1517 1519 1518 1510 1516 1516 1522 1524 1525 1525 1526	78 7A E6 7B FE ED 04 75 84 E5 24 FF ED B5 7F	29 0A 08 F0 06 F0 30	L2A: L2B:	EXIT USES MOV MOV MOV MOV MOV MOV MOV INC MOV DIV MOV ADD MOV CUNE MOV	none all DPTR, #L25MEM TEMP, #0 PTR1, #TABTAB INDX1, #10 A, @PTR1 INDX2, #8 WORK1, A A, TEMP A B, #10 AB A, B A, B A, B A, B A, #'O' WORK2, A A, TEMP A, PTR2A, L2C WORK2, #09H A, WORK1	; Counter variable ; Point to tab table ; 10 bytes ; REPEAT get byte ; 8 bits ; save value for later ; REPEAT ; get and adjust position ; calculate character ; convert to ASCII ; and save it
152D FE MOV WORK1, A ; replace new value	150D 1510 1512 1514 1516 1517 1519 1518 1510 1516 1516 1522 1524 1525 1525 1526	78 7A E6 7B FE ED 04 75 84 E5 24 FF ED B5 7F	29 0A 08 F0 06 F0 30	L2A: L2B:	EXIT USES MOV MOV MOV MOV MOV MOV INC MOV DIV MOV ADD MOV CJNE MOV	none all DPTR, #L25MEM TEMP, #0 PTR1, #TABTAB INDX1, #10 A, @PTR1 INDX2, #8 WORK1, A A, TEMP A B, #10 AB A, B A, # O' WORK2, A A, TEMP A, PTR2A, L2C WORK2, #09H A, WORK1	; Counter variable ; Point to tab table ; 10 bytes ; REPEAT set byte ; 8 bits ; save value for later ; REPEAT ; set and adjust position ; calculate character ; convert to ASCII ; and save it ; IF cursor = Position THEN ; chanse to cursor char ; restore tab table value
	150D 1510 1512 1514 1514 1516 1517 1518 1518 1516 1516 1520 1522 1524 1525 1526 1529	78 7A E6 7B FE ED 04 75 84 E5 24 FF ED B5 7F	29 0A 08 F0 06 F0 30	L2A: L2B:	EXIT USES MOV MOV MOV MOV MOV INC MOV DIV MOV ADD MOV CUNE MOV RLC	none all DPTR, #L25MEM TEMP, #0 PTR1, #TABTAB INDX1, #10 A, @PTR1 INDX2, #8 WORK1, A A, TEMP A B, #10 AB A, B A, # O' WORK2, A A, TEMP A, PTR2A, L2C WORK2, #09H A, WORK1	; Counter variable ; Point to tab table ; 10 bytes ; REPEAT set byte ; 8 bits ; save value for later ; REPEAT ; set and adjust position ; calculate character ; convert to ASCII ; and save it ; IF cursor = Position THEN ; chanse to cursor char ; restore tab table value
1530 7F 00 MOV WORK2, #00H ; change to tab ch	150D 1510 1510 1512 1514 1516 1517 1518 151B 151C 151F 1520 1522 1524 1525 1525 1526 1529	78 7A E6 FE ED O4 75 84 E5 24 FF ED B5 7F	29 0A 08 F0 06 F0 30	L2A: L2B:	EXIT USES MOV MOV MOV MOV MOV INC MOV DIV MOV ADD MOV CUNE MOV RLC	none all DPTR, #L25MEM TEMP, #0 PTR1, #TABTAB INDX1, #10 A, @PTR1 INDX2, #8 WORK1, A A, TEMP A B, #10 AB A, B A, # O' WORK2, A A, TEMP A, PTR2A, L2C WORK2, #09H A, WORK1	; Counter variable ; Point to tab table ; 10 bytes ; REPEAT set byte ; 8 bits ; save value for later ; REPEAT ; set and adjust position ; calculate character ; convert to ASCII ; and save it ; IF cursor = Position THEN ; chanse to cursor char ; restore tab table value

.....

Z29ROM: *** Z-2	MICROS 29 COMPU	ENCH 805	1 CROSS /	ASSEMBLER (\ MWARE.YERSI)	V1)-218 24-F DN 1.Q3 ***	EB-83 15:	:30:06 PAGE 187	 7 				
5 5 7 8 9	1 1533 2 1536 3 1538 4 1539 5 1538 5 1538 7 1530 3 1536 7 1537 9 1541	EF FO A3 OD DB OS	01 02 DC D5		?D :	CUNE MOV MOVX INC INC DUNZ INC DUNZ	A, PTR2A, L2I WORK2, #02H A, WORK2 @DPTR, A DPTR TEMP INDX2, L2B PTR1 INDX1, L2A		get f: Place bump f	l bits table point	.r	
		4-4- · · · · · · · · · · · · · · · · · · ·				RET		•••••••	• • • • • • • • • • • • • • • • • • • •			
••••••		• • • • • • • • • • • • • • • • • • • •	••••••					•••••				••••••
												• • • • • • • • • • • • • • • • • • • •
•••••			• • • • • • • • • • • • • • • • • • • •						•••••			
		• • • • • • • • • • • • • • • • • • • •									• • • • • • • • • • • • • • • • • • • •	
• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		•••••				• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •			
										••••••		
					• • • • • • • • • • • • • • • • • • • •							
			• • • • • • • • • • • • • • • • • • • •	•••••				• • • • • • • • • • • • • • • • • • • •				
			• • • • • • • • • • • • • • • • • • • •				• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	•••••	• • • • • • • • • • • • • • • • • • • •		
	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •				• • • • • • • • • • • • • • • • • • • •					
											••••••	•••••
	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •							
· · · · · · · · · · · · · · · · · · ·	• • • • • • • • • • • • • • • • • • • •						•••••		•••••			•••••
							•••••			•••••		
•••••									• • • • • • • • • • • • • • • • • • • •			
• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •									

(

				; ;	SETUP3	- setup mode line #3	
				*			
					SETUP	3 runs the terminal so duplex, and handshake	etup mode to chanse baud rate,
				, ;	PELICI	, udriex, and nandshak	= •
• • • • • • • • • •		•••••••		· · · · · · · · · · · · · · · · · · ·			
					ENTRY	none	
				; •	EXIT	none	
	• • • • • • • • • •						
					USES	all	
1542	Bi	98		SETUP3:	ACALL	LЗ	; Display third line
1544	. 91	07		STP3A:	ACALL	SFCKB	; Fetch character from keyboard
1546	75	FO	04		MOV	B, #L3TL	
1549 1540	<u>90</u> 71	158C E9			MOV ACALL	DPTR, #L3T LOOKUP	; Look up command
154E	80	F4			SJMP	STP3A	; Keep on going
					Change	baud rate	
: !					Chanse		
1550	E5	33		STP3B:	MOV	A, BAUDRATE	; Search for current baud rate
1552	75		9Ę		MOV	B, #L3ATL DPTR, #L3AT	
. 1555 1558	90 91	161E 04			ACALL.	XSTABX	
	40	06			JC	STP3B1	; IF found THEN
155C	E5	F9			MOV	A. B	; set count + 1
155E	B4	ΟE	01		CUNE ÇLŖ	A, #L3ATL, STP3B1 A	; IF time for wrap THEN ; Zero
15 41 2 1562	E4	0E4B		STP3B1:	CALL	SBR	; Set baud rate
1565		ДВ			\$JMP	\$ETUP3	
ļ -							
j ,				*	Change	Parity	
, , 							
1567	20	28	08	STP3C:	JB	PRTYENA, STP3C1	; IF disabled THEN
2154A) 156C	D2 C2	28 29			\$ETB	PRTYENA PRTYSTK	j enable Parity; normal parity
, 1560 154E	<u>62</u> <u>62</u>	29 29			ÇLR	PRTYEYN	, odd parity
1570	80	DO			SJMP	SETUP3	
3.1572	29	29		STP3C1:	JB	PRTYSTK, STP3C2 PRTYEVN, STP3C3	; IF normal parity AND
) 1575 5 1 578	20 Д2	2A 2A	08		SETB	PRTYEVN, STESCS	; make even parity
749.49 5.157A	80	65		***************************************	SJMP	SETUP3	
7 1 570	¢2	28		STP3C2:	CLR	PRTYENA	
157E	80 po	C2		STP3C3:	SJMP SETB	SETUP3 PRTYSTK	; IF eyen THEN stick Parity
2.158Q) 1582	D2 80	29 BE			SE.LP SJMP	SETUP3	
							
2					T1-	dumlay	
∛ }							
• 5.1584	B2	2F		ŚTP3D:	gp L	FULLDPLX	; Tossle full/balf.duplex
1586	80	BA			SJMP	SETUP3	

1								
2		•••••		* * * * * * * * * * * * * * * * * * *	Loaale	handshake	••••••	
.3. 4 1588 5 158A	B2 80	ЗЕ В6		STP3E:	CPL SJMP	HNDSHK SETUP3	; Tossle soft/hard handshake	
6 7								
.i 8				• • • • • • • • • • • • • • • • • • • •				• • • • • • •
.9 io					L3T -	line #3 table		
11							table for line #3 in setup mode.	
.3 .4	158C			·····			rable top line #2 In Seldh mode.	
.5				L3Ť	EQU	\$		
6 1580 7 1580	31 1550				DM DB	11 STP3B	; Option #1	
7 158D		• • • • • • • • • • • • • • • • • • • •		•••••			; Change baud rate	
9 158F 20 1590	32 1567				DB DW	27 STP3C	; Option #2 ; Change parity	
?1 ?2 1592	33				DB	³ 3		
23 1593	1584				DM	STP3D	; Option #3 ; Tossle duplex	
24 25 1595 26 1596	34				DB	· 4·	; Option #4	
	1588				DW DB	STP3E	; Toggle handshake	
?7 ?8	0004	• • • • • • • • • • • • • • • • • • • •		L3TL	EQU	(\$-L3T)/3	; Table length	
29 10	•••••							
3					L3 - 1	ine #3		• • • • • • •
4							·····	
5 &	• • • • • • • • • • • • • • • • • • • •				*L3*_d	isplays setup info	rmation for line #3.	
7 8	• • • • • • • • • • • • • • • • • • • •				ENTŔY	none		
9				, ,				
0 1				;	EXIT	none		
2 3		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • •	-	USES	all		• • • • • • •
4		• • • • • • • • • • • • • • • • • • • •			•••••			
5 1598 6	E4	• • • • • • • • • • • • • • • • • • • •		F3:	CLR	A	; Position	
7 1599	71	FE	 		ACALL	XSTX		
8 159B 159E	2A 4E	4D 55	45 20		DB	*MÉNU B* 1. B	AUD', (/ /+80H)	
	42 20	55 2A 20	20		•••••	• • • • • • • • • • • • • • • • • • • •	•••••	
15A1	2É	2ò	31 42					
15A4 15A7		55	44					
15A4 15A7 15AA	<u>41</u>							
15A4 15A7	41 Ao E5	33			MOV	A, BAUDRATE	***************************************	

						· · · · · · · · · · · · · · · · · · ·	
	161E			L3AT	EQU	\$	
	· · · · · · · · · · · · · · · ·				rates.		ed to determine the next.
							sses for the corresponding
				; ;	L3AT -	line #3 table	
٠	81	Q1		L3D:	AJMP	XSTATX	; Handshake message
	90	16A8			MOV	DPTR, #LN3G2	; Software handshake
13 16	90 29	16A0 3E	03		ή₿ MOΛ	DPTR, #LN3G1 HNDSHK, L3D	; Hardware handshake
o 							
	53 4B	48 45	41 AO				
:Q7 БОА	41	4E	44				
ļ 7	20 2E	20 29	34 48		DB	4. HANDSHAKE	(^ ^+80H)
Q2	7.1	FE			AÇALL.	xstx	
ωQ	21	91	• • • • • • • • • • • • • • • • • • • •	L3C:	AÇALL.	XSTATX	; Duplex message
15FD	90	169C			MOV	DPTR, #LN3E2	; Half duplex
15FA	22		03			FULLDPLX, L3C	
15F7	90	 1698			 MOV	DPTR, #LN3E1	; Full duplex
L5F4	45	58	ΑÖ				
.5EE .5F.1	2E 55	20 50	44 4C				
15EB	29	20	33		ba	3. DUPLEX:	
5E9	71	FE			ACALL	XSTX	
L5E7 	91	01		L3B:	ACALL	XSTATX	; Parity message
ξĘ4	9Ω	1493.			MQV		
5DE 5E1	<u>7</u> 9 20	168E.	 03		JB	PRTYEVN, L3B	
5DB	20 90	29	09		JB MOV	PRTYSTK, L3B DPTR, #LN3C3	; Even parity
5D8	99	1689			MOV	DPTR, #LN3C2	; Stick parity
)2)5	<u>90</u>	<u>1684</u> . 28	oF		MOV	DPTR, #LN3C1 PRTYENA, L3B	; No Parity
						gar, gar, agar gar.	
1500 1505	41 54	52 59	49 A0				
1509	2E	20 52	50 49				
1506	20	20	32		ACALL DB	XSTX 2. PARITY', ('	(+80H)
5C4	7.1	FE			ACALL	XSTX	
1502	9.1	01			ACALL.	XSTATX	; Baud rate messase.
15CO	90 74	13	• • • • • • • • • • • • •	L3A:	MOV	A, #19	; Position
15BA 15BD	75 90	33 167F	00		MOV MOV	DPTR, #LN3B14	; clear baud rate ; error messase
15B8	50	06 33			JNC	L3A BAUDRATE, #0	; IF not found THEN.
	90 91	161E			ACALL	XSTABX	; Search for correct
B3 B6	70	1015			MOV	DPTR, #L3AT	

				RE. YERSION.1.03.*			
1 10	51E	F.F			ne	255	• 19200 hand
1.10 2.10	51F	1648	• • • • • • • • • • • • • • • • • • • •		DB DW	255 LN3B	; 19200 baud
34 16		FE			<u></u>	···· <u>···</u> ···	
5.16		164D			DB DW	254 LN3B1	; 9600 baud
6							·····
7.18	24	FC 1652			DB DW	252 LN3B3	; 4800 baud
9					DW	LN3B3	
10 16	27	···FF	• • • • • • • • • • • • • • • • • • • •		DB	255	; skip 7200 baud
11 16	28	167F			DW	LN3B14	, 5 kg , 7200 Eddd
12 13 16	-20	F8			nn.	040	
14 16	2B	1657		• • • • • • • • • • • • • • • • • • • •	DB DB	248 LN3B5	; 2400 baud
15					2.4	E140E3	
16 16		``FF			DB	255	; skip 3600 baud
17 16 18	?ZE	167F			DW	LN3B14	
	30	F5			DB	245	; 1800 baud
19.16 20.16	31	F5 1650			ĎW	LN3B7	7. 1000 badd
21 22:18	'AH	···FF			DB	· · · · · <u>· · · · · · · · · · · · · · </u>	
23 16		167F			DM	255 L NOD1 4	; skip 2000 baud
24						LN3B14	
25.16 26.16	36	.FO			DB	240	; 1200 baud
26 16 27	37	1661			Ď₩	LN3B8	
28 16	39		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	DB	224	; 600 baud
29.16 30	3A	1666			₽₩	LN3B9	, coo paga
30 31 16	30	CO			DB		
32 16	3D	166B			DM	192 LN3B10	; 300 baud
33							
34 16 35 16		ිප්රි 1670			DB	128	; 150 baud
36	. 	.13/0			DW	LN3B11	
37 16	42	52			DB	82	; 110 baud
````38``16 39	43	1675			ĎW	LN3B12	
40 16	45	i			DB	····1	; 75 baud
41 16		167A			DW	LN3B13	; 75 baud
42							
43		.000E		L3ATL	EQU	(\$-L3AT)/3	; Table length
45							
46			• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	
47 48							
49				, , , , , , , , , , , , , , , , , , ,	Line #	‡3 setup mode messages	
5ō						• • • • • • • • • • • • • • • • • • • •	
51 16		31	. 39	LN3B:	DB	/1920/, (/O/+80H)	
16 52 16		30 39	B0 36 30	I NOD1	np	494004 44 4400U)	
16	50	30	AO		DB	(196001, (1 (+80H)	
53 16		.34	38 30 A0	LN3B3:	DB	/4800/, (/ /+80H)	
16 54 16		30 32					
			34 30	LN3B5:	DB	(14001, (11+80H)	

C

C

C

C

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE 191+ *** Z-29 COMPUTER TERMINAL FIRMWARE VERSION 1,03 ***
165A 30 A0

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-: *** Z-29 COMPUTER TERMINAL FIRMWARE VERSION 1.03 ***	33 15:30:06 PAGE 192
	B /1800/, (/ /+80H)
	8
1807 20 HO	8
1673 20 AO	8 /150 /, (/ /+80H)
8 167F 65 72 72 LN3B14: DI 1682 6F F2	
9 1684 73 70 61 LN3C1: DI	
10 1689 6D 61 72 LN3C2: DI 168C 6B AO	
1691 6E AO	3
12 1693 6F 64 64 LN3C4: DI 1696 20 A0	
13 1698 66 75 6C LN3E1: DE 1698 EC 14 169C 68 61 6C LN3E2: DE	
169F E6	
16A3 64 77 61	3
16A6 72 E5 16 16A8 73 6F 66 LN3G2: DE 16AB 74 77 61	
16AB 74 77 61 16AE 72 E5	······································
	······································
	······································
	······································

	-			\$ION 1,03 **	SETUP4	- setup mode line #4	
	,			7			tup mode to change port, mode
				• • • • • • • • • • • • • • • • • • • •	hold s	creen mode, and monitor	tup mode to change port, mode mode.
				7	ENTRY	none	
				;	EXIT	none	
		• • • • • • • • • • • • •		,			
					USES	all	
16B0	Di	F8		SETUP4:	ACALL	L4	; Display fourth line
16B0 16B2	91	0.7		STP4A:	CALL	SFCKB	; Fetch character from keyboard
16B4	75	FO	04		MOV	B, #L4TL	
16B7	90 71	16EC			MOV	DPTR, #L4T	I had up compand
16BA	71 80	E9 F4			CALL SJMP	LOOKUP STP4A	; Look up command ; Keep on going
16BC	8.Υ					#:!.#!	
				; ;	Loaale	main port	
16BE	12	1EOF		STP4B:	CALL	CMPORT	; Tossle port
1601	8ō	ED			SJMP	SETUP4	
						ANSI/ZDS/ADM 3/Hazelt:	ine made
				,	109916		
1603	E5	28		STP4C:	MOV	A, MODE	; Get current mode
16C5	<u>F.</u> p	<u></u>			MQ.V	TEMP, A	; Save it ; Mask off mode portion
1606 1609	53 23	05	F0		RL	A	; Rotate to next mode
16CA	22 54	OE			ANL	A, #00001110B	; Mask off extra stuff
16QC	<del>7</del> 0	01			JNZ	\$TP4C1	; IF zero THEN
16CE	04				INC	A	; bump to one
	<u>4.D</u>			STP4C1:	ORL	A, TEMP	; Combine both halfs ; Save new current mode
	F5	28 Ç <b>B</b>				TERMINIT	; Initialize for any terminal
16D0	- O.L	⊭₽ 43	02		JNB	HAZ, STP4C2	; IF Hazeltine THEN
16D0 16D2					ACALL.	HAZINIT	; initialize for Hazeltine
16D0 16D2 16D4	30 31	D.4					
16D0 16D2 16D4 16D7 16D9	30	D4 D5		STP4C2:	SJMP	SETUP4	
16D0 16D2 16D4 16D7 16D9	30 31			STP4C2:	SJMP		
16D0 16D2 16D4 16D7 16D9	30 31			STP4C2:	SJMP	SETUP4	
1600 1602 1604 1607 1609	30 31		05		Togale	.hold.screen.mode HSMODEF, STP4D1	
16D0 16D2 16D4 16D7 16D9	30 31  80  30 	D5 36 0B43	95		SJMP Toggle JNB CALL	hold screen mode HSMODEF, STP4D1 XHSM	; IF hold screen THEN exit
16D0 16D2 16D4 16D7 16D9 16D8 16DE 16DE	30 .31 .80 30 30 30	D5 36 OB43 .CD	05.	; ; STP4D:	SJMP Toggle JNB CALL SJMP	hold screen mode HSMODEF, STP4D1 XHSM SETUP4	
16D0 16D2 16D4 16D7 16D9 16D9 16DE 16DE 16E1	30 31	D5 36 OB43 CD 	05		SJMP Toggle JNB CALL SJMP CALL	hold screen mode HSMODEF, STP4D1 XHSM SETUP4 EHSM	; IF hold screen THEN exit ; ELSE enter hold screen
16D0 16D2 16D4 16D7 16D9 16D8 16D8 16D8 16E1	30 .31 .80 30 30 30	D5 36 OB43 .CD	05	; ; STP4D:	SJMP Toggle JNB CALL SJMP	hold screen mode HSMODEF, STP4D1 XHSM SETUP4	
16D0 16D2 16D4 16D7 16D9 16D8 16D8 16D8 16E1	30 31	D5 36 OB43 CD 		; ; STP4D:	SUMP Togale UNB CALL SUMP CALL SUMP	hold screen mode HSMODEF, STP4D1 XHSM SETUP4 EHSM	
16CF 16D0 16D2 16D4 16D7 16D9 16D9 16DB 16DB 16DE 16E3 16E3 16E6	30 31	D5 36 OB43 CD 		; ; STP4D:	SUMP Togale UNB CALL SUMP CALL SUMP	hold screen mode HSMODEF, STP4D1 XHSM SETUP4 EHSM SETUP4	

	4									
• • • • • • • • • • • • • • • • • • • •	ł 2		• • • • • • • • • • • • • • • • • • • •							
	آ									
	4 5				::	LAT -	line #4 table			
	6				<del>7</del> 7					
	7					*L4.T*.	contains the jume t	table.fon.line	#4.in.setur.made	
		16EC			L4T	EQU	\$			
10	0 1_16EC	21								
1:	2 16ED	31 16BE		• • • • • • • • • • •	•••••	DB DW	(1/ STP4B		le ports	• • • • • • • • • • • • • • • • • • • •
	3 <b>4 1</b> 6EF	 32		• • • • • • • • • • • • • • • • • • • •		DB				
15	5 16F0	1603				DM DB	22 STP4C	: Tnee	le mode	
10	6 7 16F2									
18	8 16F3	33 16DB			• • • • • • • • • • • • • • • • • • • •	DB DW		: Togg	le hold screen	
$\frac{19}{26}$	9 0 16F5	34				• • • • • • • • • • • • • • • • • • • •			the month of the control of the cont	· · · · · · · · · · · · · · · · · · ·
	0 16F5 1.16F6	34 16E8				DB DW	/4/ STP4E	: Toos	le monitor mode	
22	2								le.monitor.mode	
23 24	4	0004			L4TL	EQV	(\$-L4T.)/3		e.lenath	
25 26	<del>5</del>							•••••		
27						L4 - 1	ine #4	• • • • • • • • • • • • • • • • • • • •	•••••	
29 30	ž 5						isplays setup infor			
31 32						^L+^ U	setur intor	mation for [1]	He #박.	
					;	ENTOV	Dono			
33 34		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		· · · · · · <u>/</u> · · · · · · · · · · · · · · · · · · ·	ENTRY	none	• • • • • • • • • • • • • • • • • • • •		
35 36	) 					·····ÉXIT	none			
37	7				,	USES	all			
38 39								• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •
40	16F8	E4		• • • • • • • • • • • • • • • • • • • •	L4:	CLR	 A			
	16F9 2 16FB	71 2A	FE			ACALL	XSTX	• • • • • • • • • • • • • • • • • • • •		
~ <b>~</b>	16FE	4E	41) 55	45 20		DB	*MENU C* 1. PO	ORT4, (4 4+80H)	)	
	1701	43	55 2A	20 20	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	••••••	• • • • • • • • • • • • • • • • • • • •
	1704 1707	20 2É	20 20	31 50						
	170A	4F	52	54						
43	~176B~	ΑÖ								• • • • • • • • • • • • • • • • • • • •
44	170É	96	1769		• • • • • • • • • • • • • • • • • • • •	MOV	DPTR, #LN4B1	Norma	 a l	
	1711 1714	90 90	02	03		JNB	PORTF, L4A			
47	1717	90 91	176F 01		L4A:	MOV ACALL	DPTR, #LN4B2 XSTATX	; Auxil	liary	
48		• • • • • • • • • • • • •			<del></del>				message	• • • • • • • • • • • • • • • • • • • •
5ö	1719 1718	<del>71</del> 26	FE 20	<u>.</u> 20		ACALL DB	XSTX	Z. 6600	•••••	
	171E	32	. 2Ĕ	20		מט	2. MODE/, (/	+80H)		

	OM; MICROE	BENCH 80	51 CROS	s assembler ( IRMWARE VERSI	√1)-218 24 ON 1.03 ***	j-FEB-83 15: ∙	30:06 PAGE	i 194+	 				
	1724	45	#F	44					 				
***************************************									 				
	,								 			•••••	
					•••••				 				
									 				• • • • • • • • • • • • • • • • • • • •
									 				· · · · · · · · · · · · · · · ·
		,							 				
							,		 				
									 				• • • • • • • • • • • • • • • • • • • •
									 				,
				•••••					 				
									 		**************		
									 	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • •

)

)

•

)

)

1								
······································	1726	.90	1775		• • • • • • • • • • • • • • • • • • • •	MOV	DPTR, #LN4D1	
	.1722	20		OF.		JB	ANSI, L4B	; ANSI mode
	172C	90	40 177D			MÖV	DPTR, #LN4D2	; ZDS mode
5.	.172F	20	41	92		JB	ZDS, L4B	, 2D2 MOG8
	1732	90	1785			MÖV	DPTR, #LN4D3	; ADM 3 mode
	1735	20	42	03		JB	ADM3, L4B	, Apri 3 mode
	1738	90	178D			MOV	DPTR, #LN4D4	; Hazeltine mode
	.1.73B	91	01		L4B:	ACALL	XSTATX	; ANSI/ZDS/ADM 3/Haz message
10							• • • • • • • • • • • • • • • • • • • •	, individual office message
	.1Z3D	71				ACALL	XSTX	
	173F	20	20	20		ACALL.	3. HOLD SCRN/,	(/ /+80H)
	.1742	33 48	2E 4F	29				
	1745			4C				
	.1748	44	20 52	53 4E				
	174B	43	52	4E				
	.174E	<u>Ao</u>					•••••	
13	1705	Λ	2.4					
	.174F	A2	36			MOV	C, HSMODEF	
	1751	12	1BBF			LCALL	XLONOFF	; Hold on/off message
1.6.	17F4		· · · · · · · · · · · · · · · · · · ·					
	1754 1756	71	FE	20		ACALL	XSTX	
A.Y.	.#759 	29 3 <b>4</b>	29 2E	29				(+80H)
	1750			20				
	175F	4D 49	4F 54	4E	• • • • • • • • • • • • • • • • • • • •			
	1762	52	A0	41-				
19	. r		·····	• • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
	1764	A2	30			MOU	C MONITOD	
	1766	<u>A2</u>	3A 1BBF		• • • • • • • • • • • • • • • • • • • •	MOV LJMP	C, MONITOR XLONOFF	
						COM	ALONOFF	; Monitor mode on/off and return
23		• • • • • • • • • • • •		• • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			
24								
25								••••••
26 27						Line #4	l setup mode messages	
					;		,	
28	1769							
		6E	6F	72	LN4B1:	DB	'norma', ('1'+80H)	
	176C	6D	41	EC 78				
	176F	61	75		LN4B2:	DB	'auxil', (' '+80H)	
	1772	92	<u>6</u> g	<u>A0</u>	· · · · · <u>· · · · · · · · · · · · · · </u>			
	1775	61	6E		LN4D1:	DB	ansi /, (/ /+80H	)
	1778 1778	69 20	<u>20</u>	29				•••••
	177D	20 7A		45	I MATO	<b>pr. m.</b>		
	1780	/H	65 74	6E 68	LN4D2:	DB	/zenith /, (/ /+80H	)
	1783	20	/ <del>↑</del> A0	00				
	1785 1785	61	64	6p	LN4D3:		Mariana na manana na manana kaominina na manana	i
	1788	20	33		F1440-2 :	DB	'adm 3a ', (' '+80H	)
	178B	··· <del>2</del> ö·····	Ao	61	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	`	
	178D	68	61	7A	LN4D4:	DB	7557 1507 7707 000	`
	1790	<u>2</u> ŏ	31	<u>∕</u> 35	LINTLOT		'haz 150', ('0'+80H	) ·····
	1793	30	RO					
		• • • • • • • • • • • • • • • • • • • •	· · · · · <del>· ·</del> · · · · · · · ·		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
· · · · · · · · · · · · · · · · · · ·	• • • • • • • •		• • • • • • • • • • • • • • • • • • • •		•••••	• • • • • • • • • • • • • • • • • • • •		

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (VI)-218 24-FEB-83 15:30:06 PAGE 196

					SETUP5	- setup mode line #5	
				5 5	eleeof.	5* runs the terminal s status line, setup wr	etup mode to chanse set clock, ap around, and screen saver.
					ENTRY	none	
				; ;	EXIT	none	
				\$ \$	USES	all	
1795	12	1808		SETUP5:	CALL	L5	; Display fifth line
1798 1798	91 75	<u>07</u>	04	STP5A:	CALL MOV	SFCKB B, #L5TL	; Fetch character from keyboard
1790	90	17FC			MOV	DPTR, #L5T	· · · · · · · · · · · · · · · · · · ·
17A0	71	E9			CALL	LOOKUP	; Look up command
17A2	80	F4			SJMP	STP5A	5 Keep on soins
				;	Set cl	ock	
17A4 17A7	12 E4	1B2C		; STP5B:	CALL CLR	SCLRLR A	; Setup clear line reverse
		FE			ACALL	XSTX	
17A8 17AA	71 45	FE	74		DB	Enter hours 7, (7)	+80H)
17AD 17B0	65 	72 6F	20 75				
1783 1786	72 BE	73	20				
17E7	74	ÖE			MOV	A, #14	
1787 1789		1B45			CALL	GETNUM	; Get number
17BC	12 BF	00	02		CUNE	WORK2, #0, STP5B1	; Get number ; IF zero THEN skip
17BF 17C1	80 B4	D4			SUMP	SETUP5	
	B4	18	00	STP5B1: STP5B2:	CUNE UNC	A, #24, STP5B2 SETUP5	; IF number <= 24 THEN
1704 1706	50 F5	CF		311.382.	MÖV	THOUR, A	
17C8	12 E4	1B2C			CALL	SCLRLR	; Setup clear line reverse
17CB					CLR	A	
1700°	71	FE			ACALL	XSTX	
17CE	45 65	<u>6E</u>	<u>74</u>		DB	Enter minutes (*)	>/+80H)
17D1 17D4	65 GA	7.2 69	∠0 AE				
1707	6D 75	69 74	6E				
17.PA	73	29	BE				
17DD	74	10			MOV	A, #16	
17DF	74 12	1845			CALL	GETNUM	; set number
17E2 17E5	B4 50	3C	00	STP5B3:	CUNE UNC	A, #60, STP5B3 SETUP5	; IF number <= 59 THEN
17E5 17E7	50 F.5	90 30		51FJØ3:	MOV	TMIN, A	, it Hamber V= 37 IMEN
17E9	E4				CLR	^	

17EA 17EC 17EE	F5 F5	3B 3A		MOV	TSEC, A TCNT, A	; clear seconds ; and count
1.7EE	80	A5		SJMP.	SETUP5	
! !			<del>,</del>	Toeel	e 25th status line	
17F0 17F2	B2 80	37 A1	stp5C:	CPL SJMP	L250N SETUP5	; Toggle status line
				Togal	e wrap	
17F4 17F6	B2 80	28 90	STP5D:	CPL SJMP	AUTOWRAP SETUPS	; Tossle auto wrap
	• • • • • • • • • • • • • • • • • • • •		; ;	Tossì	e screen saver mode	
17F8 17FA	B2 80	3B 99	STP5E:	CPL SJMP	SCRNSAVE SETUP5	; Toggle screen saver
	• • • • • • • • • • • • • • • • • • • •		;;		line #5 table	
		•••••	······		contains the jump t	able for line #5 in setup mode.
17FC	17FC		L5T	EġÜ	\$	
17FC 17FD	31 17A4			DM DB	'1' STP5B	; Set clock
17FF 1800	32 17F0	•••••		DB DW	121 STP50	; Toggle status line
i802	33			pB	,3,	, rossie status ille
1803	17F4			DW	STP5D	; Toggle wrap around
1805 1806	34 17F8			DB DW	STP5E	; Toggle screen saver
••••••	0004		L5TL	EQU	(\$-L5T)/3	; Table length
• • • • • • • • • • • • • • • • • • • •				L5 -	line #5	
• • • • • • • • • • • • • • • • • • • •				*L5* ·	displays setup infor	mation for line #5.
		• • • • • • • • • • • • • • • • • • • •	· · · · · · · · · · · · · · · · · · ·	ĖNTRÝ	none	
				EXIT	none	

.....

C

C

( (

1						USES	all				
$\dots$ $\frac{1}{2}$							•••••	• • • • • • • • • • • • • • • • • • • •			
3	1808	E4			L5:	clr	A				
5	1000				23.	OLIN	**				
	1809	<del>,</del> 71	àp			ACALL	XST				
7.	180B	2A 4È	4D 55	<mark>45</mark>		DB	*MENU D*	1. SET C	LOCK	2. STATUS LINE	, (/ /+80H)
	180E'''	4E 44	55 2A	20 20							
	1814	···2ö ····	2Ծ	3i							
	1817	2E	20	53							***************************************
	TEIA	45 43	54 4C	20 4F							
	181D 1820	···43····	4b	20							
	1823	20	20	32							
	928	5E	50	<del>5</del> 3							
	1827 1820	54 55	53	54 20							
	182C 182F	4C	53 <b>4</b> 9	4E							
		···45	Aó								
8											
	1834	A2	37			MOV	CCT L250N T XLONOFF			Status line on/o	.ee
10 11	.1836	12	1BBF			LCALL				Status Tine 0070	
	1839	71	9D 2∂			ACALL DB	XST				
	183B	71 20	26	20		p.g.	3. WRA	₽2,11(2112∓8)	OH)		
	183E	33 57	2E 	<u>20</u>							
	1841	50 50	A0	41							
14	1844			• • • • • • • • •							
15	1846 1848	<u>A2</u>	2B 188F			MOV	C, AUTOWRA	P		. , ,	.,,
		12	1BBF			LCALL	XLÓNÓFF		,	Wrap around on/o	++
17		<del>7</del> 1	9b			ACALL	xst				
	184D	20	20 2È	20 20		DB		EEN SAVER1	, (1.1	(+80H)	
	1850	34	2É	20							
	1853 1856	53 45	43 45	52 4É							
	1859	20	53	41							
	. 1859 1850	<u>20</u> 56	53. 45	41 52							
<u>2</u> 0	.185F	A0									
20	1860	A2	38			MOV	C, SCRNSAV	E			
···· <del>2</del> 2	.1860 	<u>A2</u>	3B 1BBF	• • • • • • • • • •		LJMP	XLONOFF	T	· · · · · · · · · · · · · · · · · · ·	Screen saver on/	off and return
				<b></b> .							
			• • • • • • • • • • • • • • • • • • • •								
											••••••••••

				ŞETUP6	k.π.setup.mode.line	÷.#6
			;			
	• • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	;	#ŞĘJ.Vt keypad	14%.CUDS.the.tecmic 1 alternate, auto r	al.setup.mode.to.chanse.keypad.shift, epeat, and key click.
						••••••••••••••••••••••••
			; 	ENTRY	none	
			;			
			· · · · · · · · · · · · · · · · · · ·		none	
					a].l	
1865	11	97	SETUP6:	ACALL	L6	; Display sixth line
.1867 186A	<u>12</u> 75	1407 F0 04	STP6A:	ÇAĻL MOV	SFCKB B, #L6TL	; Fetch character from keyboard
186D	90 12	188B		MOV	DPTR, #LAT	
1870	12 80	13E9		CALL	LOOKUP	; Look up command
. 1873	8	F.2		SUMP	STP.6A	, Keep on acina
		• • • • • • • • • • • • • • • • • • • •				
			······································	109916	e keypad shifted	
1875	B2	30	STP6B:	CPL	KPADSHTF	; Toggle keypad shifted
.1877	¤y	EÇ	• • • • • • • • • • • • • • • • • • • •	SJMP	SETUP6	
			; :	Loaale	keypad alternate	
1879	B2	31	STP6C:	CPL	KPADALTF	; Toggle kerpad alternate
.197B	89	E8		SJMP	SETUP6	
				•••••		
			; ;		e auto repeat	
187D	B2	34	STP6D:	CPL	REPF	; Tossle auto repeat
187F 1882	<u>12</u>	119E E1		CALL SJ <b>M</b> P	INITKB SETUP6	; Reconfigure keyboard
		<del>_</del>				
			•	Tonnlo	key click	
		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • •			••••••
1884 1886	B2 12	33 119E	STP6E:	CPL CALL	CLKF INITKB	; Togale key click
1889	80	DA		SJMP	SETUP6	; Reconfigur keyboard
					•••••	
		••••••			• • • • • • • • • • • • • • • • • • • •	
		• • • • • • • • • • • • • • • • • • • •			,	
			; ;	L6T -	line #6 table	
			•••••••	*L6T*	contains the jump	table for line #6 in setup mode.
	188B	• • • • • • • • • • • • • • • • • • • •	L6T	EQU	\$	
	•		LO1			
188B	31			DB DW	STP6B	·····

1							
2 188E	32				DB DW	'2' STP6C	. Togglo kovpad alternate
3 188F 4	1879						; Toggle keypad alternate
5 1891	33 187D				DB DW	/3/ STP&D	
6 1892 7							; Tossle auto repeat
8 1894	34		• • • • • • • • • • • • • • • • • • • •		DB		
9 1895 0	1884				DW	STP6E	; Toggle key click
i 2	0004		L6	TL	EQU	(\$-L6T)/3	; Table length
3							
4			• • • • • • • • • • • • • • • • • • • •				
5						ine #4	
7			;				
8					*L6* d	isplays setup info	rmation for line #6
:ö							
:i :2					ENTRY	none	
:2 :3			; ;		EXIT	none	
:3 :4			· · · · · · · · · · · · · · · · · · ·				
:5 :6					USES	all	· · · · · · · · · · · · · · · · · · ·
?7							
8 1897 9	E4		L6	•	CLR	A	; Position
.7 80 1898	7i	9D	• • • • • • • • • • • • • • • • • • • •		ACALL	XST	
1 189A	2A 4E	4D 55	<b>45</b> 20		DB	/*MENU E* 1. k	EYPAD SHFT', (' '+80H)
189D 18A0	45	2A	20				
18A3	20	20	31				
18A6 18A9	2E λ≒	<u>20</u>	4B 50				
18AC	2E 45 41	44 48	20				
18AF	53 54	48 A0	46				
18 <b>B</b> 2							
3 18 <b>84</b> 4 18 <b>8</b> 6	A2	30			MOV ACALL	C, KPADSHTF XLONOFF	• Mayand shifted as /aff
14 1280 15							; Keypad shifted on/off
5 6 1888	71	9D			ACALL	XST	T/ // /.00U
18BA 18BD	20 32	20 2É	20 20		DB	1 2. KEYPAD AL	T/, (/ /+80H)
1800	4B 50	45 41	59 44				
1803	50	41	44 40				
1806 1809	20 54	41 AO	46				
38							
9 18CB 10 18CD	A2 71	31 BF			MOV ACALL	C, KPADALTF XLONOFF	; Kerpad alternate on/off
1							
12 18CF 13 18D1	<u>71</u>	9D 20	20		ACALL	XST 3. REPEAT(,	(< <+90H)
18D1 18D4	20 33	20 2E	20		מע	S. REFERI	7 (0011)

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE 200+ *** Z-29 COMPUTER TERMINAL FIRMWARE VERSION 1.03 ***
18D7 52 45 50 18DA 45 41 54
18DD AO
······································
······································

229ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-F *** Z-29 COMPUTER TERMINAL FIRMWARE VERSION 1.03 ***	-EB-83 15:	:30:06 PAGE 201
1 2 18DE A2 34	MÖV	C, REPF XLONOFF ; Auto repeat on/off
3 18E0 71 BF		
5 18E2 71 9D 6 18E4 20 20 20 18E7 34 2E 20	DB DB	XST 4. CLICK?, (? ?+80H)
18E7 34 2E 20 18EA 43 4C 49 18ED 43 4B AO		
18ED 43 4B AO		O CIVE
7 8 18F0 A2 33 9 18F2 61 BF	MOV AJMÞ	C, CLKF  XLONOFF  3 Key click on/off and return
	• • • • • • • • • • • • • • • • • • • •	
	• • • • • • • • • • • • • • • • • • • •	

	1				; ;	SETUP7	- setup mode line	#7
	2		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	*			
• • • • • • • • • • •	ই	• • • • • • • • • • • • • • • • • • • •				SETUP.	7* runs the termina - auto line feed, c	l.setup.mode.to.chanse.auto.carriage
	5							
	6 7				;	ENTRY	nono	
• • • • • • • • • • • •	.;		• • • • • • • • • • • • • • • • • • • •		***************************************	ENINT	ione	
• • • • • • • • • • • • • • • • • • • •	9 10					EXIT	none	•••••
	11				, ,	USES	all	
	12 13							
	13 14 18F4	эi	ЗF	• • • • • • • • • • • • • • • • • • • •	SETUP7:	ACALL	L7	; Display seventh line
	15 18F6	<u>12</u> 75	1407		STP7A:	CALL	SFCKB	; Fetch character from keyboard
	16 18F9 17 18FC	90	F0 1933	04		MOV	B, #L7TL DPTR, #L7T	
	18 18FF	12	1933 13E9			CALL	LOOKUP	; Look up command
	191902 20	80	F2			SUMP	STP7A	; Keep on soins
	21 22							
	 22	• • • • • • • • • • •			····	Toggle	auto carriase retu	rn
• • • • • • • • • • •	23 24 1904		2C	• • • • • • • • • • • • • • • • • • • •	; 	CPL	AUTOCR	; Toggle auto CR
	25 1906 26	80	EC			SJMP	SETUP7	, lossie adto ch
	26 27							
	28		• • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	· · · · · · · · · · · · · · · · · · ·	Loaale	auto line feed	
• • • • • • • • • • • • • • • • • • • •	29 30 1968 -	<u>.</u> B2	2D		.; 	CPL	AUTOLF	
:	31 190A	80	E8		31770.	SJMP	SETUP7	; Toggle auto LF
	32 33			•••••				
	34		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	· · · · · · · · · · · · · · · · · · ·	t	cursor	
	35 36 1900				; 			
:	37 190F	20 20	38 32	ОЙ О4	STP7D:	JB 	CRNBK, STP7D2 CRUL, STP7D1	; IF blinking AND ; IF block THEN
:	38 1912	D2	32 32		• • • • • • • • • • • • • • • • • • • •	SETB	CRUL	; set to blink line
	39 1914 40	80	11			SJMP	STP7D4	
	41 1916	C2	32 38		STP7D1:	CLR	CRUL.	; IF blink line THEN
	42 1918 . 43 191A	D2 80	38 0B	• • • • • • • • • • • • • • • • • • • •		SETB	CRNBK	; set to block
••••••	43 191A 44					SJMP	STP7D4	
	45 191C	20	32 32	04	STP7D2:	JB	CRUL, STP7D3	; IF block THEN
	46 191F 47 1921	D2 80	32 04			SETB SJMP	CRUL STP7D4	; IF block THEN ; set to underline
	48		• • • • • • • • • • • • • • • • • • • •					
	49 1923 50 1925	C2 	32 38		STP7D3:	CLR	CRUL	; IF underline THEN
	51		<b>ು</b> ರ			CLR	CRNBK	; set to blink block
	5211927	12	117B		STP7D4:	CALL	SETCF	; Set cursor format
	53 192A 54	80	cs			SUMP	SETUP7	
	55		• • • • • • • • • • • • • • • • • • • •					
	56 57				;	Loaale	line frequency	•••••••••••••••••••••••••••••••••••••••

1920	B2 12	39 1178		STP7E:	CPL	FREQ	; Toggle frequency
192E					CALL	SETCF	; Set cursor format
. 1931	80	C1			SJMP	SETUP7	
				3.4	L7T÷.	line #7 fable	
						Containe the limb	table for line #7 in setup mode.
							table for fine w/ in Setal model
	1933			L7T	EØN	\$	
1933	31			• • • • • • • • • • • • • • • • • • • •			
1934	1904				DW	STP7B	; Toggle auto carriage return
1936	32				DB	/2/	
1937	1908				pm	STP7C	; Toggle auto line feed
1939	33				pa	,3,	
193A	190C				D₩	STP7D	; Togale cursor
1930	34				DB	444	
193D	```192C``				DA DB	STP7E	; Togale line freg 50/60 Hz
	····oʊoʊ-			<u>L</u> ††L	E0U	('≸-ĽŹŤ)/3	; Table lensth
					L7 - 1	ine #7	
					*L7* c	lisplays setup info	ormation for line #7
				; :			
					····ÉNTRÝ	none	
					EXIT	none	
				, , , , , , , , , , , , , , , , , , ,			
				5	ÜSES	all	
193F	E4			L7:	CLR	A	; Position
1940	71 2A	9D 4D			ACALL.	XST	
1942	2A		45		DB	*MENU F* 1. 4	AUTO CR/, (/ /+80H)
1945 1948	4E 46	55 2A	20 20				
194B	20	20 20	31 41				
194E 1951	2E 55	20 54	41 4F				
i o E A	55 20	54 43	4F 52				
1957	A0						
1958 195A	A2	2C BF			MOV	C, AUTOCR	
1950	71	BF			ACALL	XLONOFF	; Auto CR on/off

	C. M911. S	41 E.D. 17E.C	ለመንግስዛሮ ነው ነን	RUMHUZE	/ERSION.1.03.*	**			
1	1950	71	90			ACAL I	VCT		
·····ž·	.195C 195E	20	9D 20	20	• • • • • • • • • • • • • • • • • • • •	AÇALL DB	0.51	AUTO LF', ('	
	1961	32	2F	20		DD	4.	HOTO EF 5 (	+00H/
	1964	41	2E 55	<del>5</del> 9	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • •		
	1967	4F	20	4C					
	196A	46	A0		• • • • • • • • • • • • • • • • • • • •				•••••••••••••••••••••••••••••••••••••••
3									
	196C	A2	2D		• • • • • • • • • • • • • • • • • • • •	MOV	C, AUT	7) F	
	196E	71	BF			ACALL	XLONOF		; Auto LF on/off
6								• • • • • • • • • • • • • • • • • • • •	7 Haco El 6117011
7	1970	71	9D			ACALL	<u>X</u> ST		
8.	1972	20	9D 20	20		ACALL		CURSORY, (272	+80H)
	1975	33	2E 55	20 52					
	1978	43	55						
	197B	53	4F	52					
	197E	ΑÖ						• • • • • • • • • • • • • • • • • • • •	
9									
	197F	20	38	ÖB		JB	CRNBK	L7C1	
	1982	90	1987			MOV	DPTR, #	#LN7D1	; Blink block
	1985	30	32	ÖË		JNB	CRUL, L	_7¢2	
	1988	90	1900			MOV	DPTR, #	\$LN7D2	; Blink underline
	198B	80	09			SUMP	``L7¢2```		•••••••••••••••••••••••••••••••••••••••
	198D	90	1909 32		L7C1:	MOV	DPTR, #		; Block steady
	1990	30		03		JNB	CRUL, L	.7C2	
	1993	90	19D2			VOM	DPTR, #	\$LN7D4	; Underline steady
	1996	71	A8		L7C2:	ACALL	XSTAT		; Cursor message
19		<u></u>							
	1998	71	9Ď			ACALL	XST		
21	199A	20	20 2É	20 20		DB	4.	FREQ1, (1 1+80	OH)
	199D	34	2E						
	19A0 19A3	<u>46</u>	52 Ao	45					
22	1783	21	AU						
	19A5	90	19BB				n nek eksemen nin na		·····
	19A8	30	39	03			DPTR, #		; 50 Hz
	19AB	80	19DC	03		JNB	FREQ, L		***************************************
	19AE	71			L7D:	MOV	DPTR; #	FLN/F2	; 60 Hz
27.		· · · · · · · · · · · · · · · · · · ·	A8		L/U•	ACALL	XSTAT		
	19B0	71	on.			ACALL	VCT		
	1374474111	····áð·····	9D 20	48		DB	XST	72121880000000	
	1982 1985	FA	20	70		סט	0 m ,	(/z/+80H)	
.39			• • • • • • • • • • • • • • • • • • • •						
31	19B6	22				RET			
32			• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • •		
.33 .34	• • • • • • • •	· · · · · · · · · · · · · · · · · · ·	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • •		
35									
.36.		• • • • • • • • • •				Line #7	setup m	ode messages	
37					;	""			
38	••••••	• • • • • • • • • •				• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	••••••
	19B7	62	6C	6E	LN7D1:	DB	/b]nk h	olc1, (1k1+80H)	)
	198A	9B	6C 20	6E	• • • • • • • • • • • • • • • • • • • •				
	19BD	6C		EB					
	1900	62	63	6Ë	LN7D2:	ĎB	751nk 1	in', ('e'+80H)	······································
	1903	6B	20				• • • •		
	1906	6B	20 6E	6C E5	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	***************************************
/11	1909	62	6C	6F	LN7D3:	DB	41.3	/, (/ /+80H)	

**C**/

C

 $\mathbf{C}$ 

Z29ROM; MI *** Z-29 C	CROBENCH	8051 CROS TERMINAL F	S ASSEMBLER IRMWARE VER	(V1)-218 KSION 1,03 *	24-FEB-83 15: ***	30:06 PAGE 2	204+	 ,		
19 19								 	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
<i>x.</i>										
			•••••					 		
• • • • • • • • • • • • • • • • • • • •								 	• • • • • • • • • • • • • • • • • • • •	
								 	• • • • • • • • • • • • • • • • • • • •	
								 		• • • • • • • • • • • • • • • • • • • •
								 		• • • • • • • • • • • • • • • • • • • •
								 		•••••
		• • • • • • • • • • • • • • • • • • • •						 	••••	
								 	• • • • • • • • • • • • • • • • • • • •	
	• • • • • • • • • • • • • • • • • • • •							 	• • • • • • • • • • • • • • • • • • • •	
							,	 		
						,		 		
		• • • • • • • • • • • • • • • • • • •						 		• • • • • • • • • • • • • • • • • • • •

Z29 ***	ROM; MIC Z-29 CC	ROBEN MPUTE	CH 8051 R TERMI	CROSS NAL FIR	ASSEMBLE MWARE VE	ER (V1)-2 ERSION 1.	218 24-F 03 ***	EB-83 15	5:30:08	PAGE 20								• • • • • • • • • • • • • • • • • • • •	
	1 19I 19I	)2 )5 )e	75 65 49	6E 72	64 60	LN7D4: LN7F1: LN7F2:		DB	(97)	lerliní.	.(1e1+80	)H)			•••••	•••••			
	191 2 191 3 191	)B )C	B5 B6			LN7F1: LN7F2:		DB DB	(757 (161	+80H) +80H)									
	• • • • • • • • • • • • • • • • • • • •											• • • • • • • • • • • • • • • • • • • •							
								•••••			• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •					• • • • • • • • • • • • • • • • • • • •	• • • • •
			• • • • • • • • • • • • • • • • • • • •							• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •			
					•••••	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •						
					• • • • • • • • • • • • • • • • • • • •														
			• • • • • • • • • • • • • • • • • • • •				••••••												
																			••••
						• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	••••••						• • • • • • • • • • • • • • • • • • • •			••••
		• • • • • • • • • • • • • • • • • • • •																•••••	••••
	• • • • • • • • • • • • • • • • • • • •	•••••	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •					• • • • • • • • • • • • • • • • • • • •										••••
		• • • • • • • • •						• • • • • • • • • • • • • • • • • • • •											
		• • • • • • • • •		•••••	• • • • • • • • • • • • • • • • • • • •						• • • • • • • • • •							• • • • • • • • • • • • • • • • • • • •	••••
						• • • • • • • • • • • • • • • • • • • •						• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •					
		• • • • • • • •								• • • • • • • • • • • • • • • • • • • •							• • • • • • • • • • • • • • • • • • • •		
	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • •		•••••		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •											••••

C

C

(

1				; ;	SETUP8	- setup mode line #8	
2				<b>5</b>			setup mode for simple terminal
3					diaano: *\$E!UP	B* runs the terminal	Setup mode for Simple terminal
5							
6				;			
<u>z</u>					ENTRY	none	
8 9				, ,	EX.IT	none	
10				;			
.11					USES	all	
12							
.13 14 19DD	51	8C		SETUP8:	ACALL	L8	; Display eighth line
.15.19DE		1.49.7		STP8A:	CALL	SFCKB	; Fetch character from keyboard
16 19E2	75	FO	04		MOV	B, #L8TL	
.17.19E5		1A74			MOV CALL	DPTR, #L8T LOOKUP	; Look up command
18 19E8		13E9 F2			SJMP	STP8A	; Keep on going
.12.1255 20	·						
.21							
22				;	Fill s	creen	
.23 24 19EI	 71	2C		 STP8B:	ACALL	SCLRLR	; Setup clear line reverse
.25.19EB	E4				CLR	A	; Position
26						VOT	
.27.19E0		9.0			ACALL DB	XST 'Press RETURN to ex	i', ('t'+80H)
28 19F2 19F5		72 73	65 20		DB	1, ess NE10111 to ex	
19F8		45	54				
19FI	3 55	52	4E				
19F6		74	6F 78				
1A0		65 F4	/.8				
.29						,	
30 1A0	74	45			MOV	A, #'E'	; Display 'E' first time
.31.1A08		500.0		STP8B1:	MQV ACALL	DPTR. #CHARMEM SFILL	; Point to character memory ; Fill screen with char
32 1A01		21			HUHLL		7 - 4 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1
34 1AO		1407			CALL	SFCKB	; Fetch character from keyboard
.35.1A1	)	F.O	0.4		V.DM	B, #LSTL2	
36 1A13	3 90	1A80			MOV	DPTR, #L8T2 STAB	; Search the table
.371A1 38 1A1		Ω13A ED	• • • • • • • • • • • • • • • • • • • •		ÇAĻĻ JC	элня STP8B1	; IF not found THEN display it
38 1A1 391A1		ED			CLR	A	
40 1A1					JMP	@A+DPTR	; Jump to routine
. <b>41</b>						Α, #΄΄	; Chanse funny space to real
42 1A1		20 E7		STP8B3:	MOV SJMP	A, #' ' STP8B1	; Fill screen
43.1A1	<del>.</del> జం					, as 11. was *	
. 45							
46				;	Chanse	attribute	
4.7				;		A, ATTRIBUTES	; Get current attributes
48 1A2	1 E5 304	21		01F0U•	INC		; Bump to next attribute
50 1A2		oF			ANL	A, #00001111B	; Mask just attributes
	53		EO		ANL	ATTRIBUTES, #111100	000B ; Get attributes ready

0.4000	42	21		QRL	ATIRIBUTES, A	; Move new attributes in
2 1A2B	90	4000		MOV	DPTR, #ATRIBMEM	; Point to attribute memory
.3.1 <b>.</b> 2.	7.1			AÇALL.	SFILL	
4 1A30 5	80	AB		SJMP	SETUP8	
. ×		• • • • • • • • • • • • • • • • • • • •	***************************************			•••••
. <u>7</u>	· • • • • • • • • • • • • • • • • • • •			Togale	echaracter.set	
8 .9.1832	na	on	;			
.Z.AMS4 LO 1A34	B2 A2	3D	STP8D:	CPL	ACHR2	; Toggle character set
11 1A36				MOV MOV	C, ACHR2 ALTCHARA, C	
12 1A38	92 80	oc	•••••	SJMP	MF.190489 SETUP8	; Place into attribute
ış	· • · • • • • • • • • • • • • • • • • •					
l 4 l 5			*	O		
!∺  6		• • • • • • • • • • • • • • • • • • • •	***************************************		nuous test	
17.1A3A	75	40	00 STP8E:	MQ.Y	ERRORS, #0	
8 1A3D	D1	D9		ACALL	ROMCHK	t check ROM
9.1A3F	D1	F1		AÇALL.	RAMCHK	
20 1A41	12	OBAC	STP8E0:	CALL	CLRS	; clear screen
?11A44	71	29		AÇALL.	\$QLRLR	i setum olean line revense
22 <b>1A4</b> 6 23	E4			CLR	A	; Position
64 24.1A47	71	9D	•••••	ACALL	xst	
25 1A49	43	6F	¢E	DB		(†/+80H)
1A4C	74	69	6E	· · · · · · · · · · · · · · · · · · ·		. St. 189117
1.A4F	75	6F	75			
1A52	73	20	74			
1A55	65	73	F4			
.,		• • • • • • • <del>• •</del> • • • • • •				
1A52 1A55 26						
7.1A58	12	02B6		CALL	FCKB	i Get any char from keyboard
7.1 <u>458</u> 8 1458	12 50	02 <b>₽</b> 6		JNC	SETUP8	; IF any char THEN exit
7.1A58	12	02B6			FÇKB SETUP8 RING	; Get any char from keyboard ; IF any char THEN exit ; Ring bell
7 1458 8 1458 9 145D 0 1460	12 50 12 E4	02 <b>₽</b> 6		JNC CALL	SETUPS RING	; IF any char THEN exit
7 1458 8 1458 9 1450 0 1 1460 2 1461	12 50 12 E4 90	0286 80 0668 5000	STP8E1:	JNC	SETUPS RING A	; IF any char THEN exit ; Ring bell
7 1458 8 1458 9 1450 0 1 1460 2 1461 3 1464	12 50 12 E4 90	0286 80 0668 5000		JNC CALĻ CLR	SETUPS RING A DPTR, #CHARMEM	; IF any char THEN exit ; Ring bell ; REPEAT point to char memory
7 1458 8 1458 9 1450 0 1 1460 2 1461 3 1464 4 1466	12 50 12 E4 90 71	0286 80 0668 5000 21 0268		JNC CALL CLR MOV ACALL CALL	SETUPS RING A DPTR, #CHARMEM SFILL LKB	; IF any char THEN exit ; Ring bell ; REPEAT Point to char memory ; fill with char
7, 1458 18 1458 19 1450 0 1 1460 2 1461 3 1464 4 1466 5 1469	12 50 12 	0286 80 0668 5000 21		JNC CALL CLR MOV ACALL CALL JNC	SETUPS RING A DPTR, #CHARMEM SFILL	; IF any char THEN exit ; Ring bell ; REPEAT Point to char memory ; fill with char ; look at keyboard
7, 1458 8, 1458 9, 1450 0, 1, 1460 1, 1461 3, 1464 4, 1466 5, 1469 6, 1468	12 50 12 E4 90 71 12 50	0286 80 0668 5000 21 0288 D6		JNC CALL CLR MOV ACALL CALL JNC INC	SETUPS RING A DPTR, #CHARMEM SFILL LKB STPSEO A	; IF any char THEN exit ; Ring bell  ; REPEAT point to char memory ; fill with char ; look at keyboard ; IF any char THEN exit ; bump to next char
7 1458 9 1458 9 1450 10 1 1460 2 1461 3 1464 4 1466 5 1469 6 1468 7 1460	12 50 12 E4 90 71 12 50	0286 80 0668 5000 21 0288 D6		JNC CALL CLR MOV ACALL CALL JNC INC JNZ	SETUPS RING A DPTR, #CHARMEM SFILL LKB STPSEO A STPSE1	; IF any char THEN exit ; Ring bell ; REPEAT point to char memory ; fill with char ; look at keyboard ; IF any char THEN exit
7. 1458 8. 1458 9. 1450 0. 1. 1460 2. 1461 3. 1464 4. 1466 5. 1469 6. 1468 7. 1466 8. 1466	.12. 50 12. .E4. 90 71. 12. 50. 04. 70. E5.	0286 80 0668 5000 21 0268 D6 F3		JNC CALL CLR MOV ACALL CALL JNC INC JNZ MOV	SETUPS RING  A DPTR, #CHARMEM SFILL LKB STPSEO A STPSE1 A, ERRORS	; IF any char THEN exit ; Ring bell ; REPEAT point to char memory ; fill with char ; look at keyboard ; IF any char THEN exit ; bump to next char ; UNTIL all combinations
// 1458 // 1458 // 1450 // 1460 // 1460 // 1464 // 1466 //	.12. 50 12. .E4. 90 71. 12. 50. 04. 70. E5.	0286 80 0668 5000 21 0268 D6 F3		JNC CALL MOV ACALL CALL JNC INC JNC JNC JNZ MOV JZ	SETUPS RING  A DPTR, #CHARMEM SFILL LKB STPSEO A STPSE1 A, ERRORS STPSE	; IF any char THEN exit ; Ring bell  ; REPEAT Point to char memory ; fill with char ; look at keyboard ; IF any char THEN exit ; bump to next char ; UNTIL all combinations ; UNTIL errors
7. 1458 7. 1450 10. 1460 11. 1460 12. 1461 13. 1464 14. 1466 14. 1466	12 50 12 E4 90 71 12 50	0286 80 0668 5000 21 0288 D6		JNC CALL CLR MOV ACALL CALL JNC INC JNZ MOV	SETUPS RING  A DPTR, #CHARMEM SFILL LKB STPSEO A STPSE1 A, ERRORS	; IF any char THEN exit ; Ring bell ; REPEAT point to char memory ; fill with char ; look at keyboard ; IF any char THEN exit ; bump to next char ; UNTIL all combinations
7. 1458 9. 1450 0. 1. 1460 2. 1461 3. 1464 4. 1466 5. 1469 6. 1468 7. 1460 8. 1466 8. 1466 9. 1472 1. 1472	.12. 50 12. .E4. 90 71. 12. 50. 04. 70. E5.	0286 80 0668 5000 21 0268 D6 F3		JNC CALL MOV ACALL CALL JNC INC JNC JNC JNZ MOV JZ	SETUPS RING  A DPTR, #CHARMEM SFILL LKB STPSEO A STPSE1 A, ERRORS STPSE	; IF any char THEN exit ; Ring bell  ; REPEAT Point to char memory ; fill with char ; look at keyboard ; IF any char THEN exit ; bump to next char ; UNTIL all combinations ; UNTIL errors
// 1458 // 1458 // 1450 // 1450 // 1460 // 1460 // 1464 // 1466 //	.12. 50 12. .E4. 90 71. 12. 50. 04. 70. E5.	0286 80 0668 5000 21 0268 D6 F3		JNC CALL MOV ACALL CALL JNC INC JNC JNC JNZ MOV JZ	SETUPS RING  A DPTR, #CHARMEM SFILL LKB STPSEO A STPSE1 A, ERRORS STPSE	; IF any char THEN exit ; Ring bell  ; REPEAT Point to char memory ; fill with char ; look at keyboard ; IF any char THEN exit ; bump to next char ; UNTIL all combinations ; UNTIL errors
7. 1458 7. 1458 9. 1450 0. 11460 2. 1461 3. 1464 4. 1466 5. 1469 6. 1468 7. 1466 7. 1466 9. 1470 0. 1472 1. 22	.12. 50 12. .E4. 90 71. 12. 50. 04. 70. E5.	0286 80 0668 5000 21 0268 D6 F3		JNC CALL CLR MOV ACALL CALL JNC INC JNZ MOV JZ AJMP	SETUPS RING A DPTR, #CHARMEM SFILL LKB STPSEO A STPSEI A, ERRORS STPSE ERROR	; IF any char THEN exit ; Ring bell  ; REPEAT Point to char memory ; fill with char ; look at keyboard ; IF any char THEN exit ; bump to next char ; UNTIL all combinations ; UNTIL errors
7. 1458 9. 1450 0. 1460 2. 1461 3. 1464 4. 1466 5. 1469 6. 1468 7. 1460 9. 1470 0. 1472 1.	.12. 50 12. .E4. 90 71. 12. 50. 04. 70. E5.	0286 80 0668 5000 21 0268 D6 F3		JNC CALL CLR MOV ACALL CALL JNC INC JNZ MOV JZ AJMP	SETUPS RING  A DPTR, #CHARMEM SFILL LKB STPSEO A STPSE1 A, ERRORS STPSE	; IF any char THEN exit ; Ring bell  ; REPEAT Point to char memory ; fill with char ; look at keyboard ; IF any char THEN exit ; bump to next char ; UNTIL all combinations ; UNTIL errors
7. 1458 9. 1450 0. 1460 2. 1461 3. 1464 4. 1466 5. 1469 6. 1466 8. 1466 8. 1466 9. 1470 0. 1472 1.	.12. 50 12. .E4. 90 71. 12. 50. 04. 70. E5.	0286 80 0668 5000 21 0268 D6 F3	STP8E1:	JNC CALL CALL CALL JNC INC JNZ MOV JZ AJMP	SETUPS RING A DPTR, #CHARMEM SFILL LKB STPSEO A STPSEI A, ERRORS STPSE ERROR	; IF any char THEN exit ; Rins bel]  ; REPEAT point to char memory ; fill with char ; look at keyboard ; IF any char THEN exit ; bump to next char ; UNTIL all combinations ; UNTIL errors ; Jump to error routine
7. 1458 7. 1458 9. 1450 0. 11460 2. 1461 3. 1464 4. 1466 5. 1469 6. 1466 7. 1460 8. 1460 9. 1470 0. 1472 1.	12 50 12 E4 90 71 12 50 04 70 E5 60	0286 80 0668 5000 21 0268 D6 F3	STPSE1:	JNC CALL CALL CALL JNC INC JNZ MOV JZ AJMP	SETUPS RING A DPTR, #CHARMEM SFILL LKB STPSEO A STPSEI A, ERRORS STPSE ERROR	; IF any char THEN exit ; Ring bel]  ; REPEAT point to char memory ; fill with char ; look at keyboard ; IF any char THEN exit ; bump to next char ; UNTIL all combinations ; UNTIL errors ; Jump to error routine
7. 1458 7. 1458 9. 1450 0. 11 1460 2. 1461 3. 1464 4. 1466 5. 1469 6. 1468 7. 1466 7. 1466 9. 1470 0. 1472 1. 22 3. 44 5. 66 7. 7. 88 6. 88 6. 88 6. 88 6. 88 6. 88 7.	.12. 50 12. .E4. 90 71. 12. 50. 04. 70. E5.	0286 80 0668 5000 21 0268 D6 F3	STP8E1:	JNC CALL CALL CALL JNC INC JNZ MOV JZ AJMP	SETUPS RING  A DPTR, #CHARMEM SFILL LKB STPSEO A STPSEI A, ERRORS STPSE ERROR  line #8 table contains the jump tabl	; IF any char THEN exit ; Rins bel]  ; REPEAT point to char memory ; fill with char ; look at keyboard ; IF any char THEN exit ; bump to next char ; UNTIL all combinations ; UNTIL errors ; Jump to error routine
1. 1458 1. 1450 1. 1450 1. 1460 1. 1461 1. 1464 1. 1464 1. 1466 1.	12 50 12 E4 90 71 12 50 04 70 E5 60	0286 80 0668 5000 21 0268 D6 F3	STPSE1:	JNC CALL CALL CALL UNC INC JNZ MOV JZ AJMP  LST - *LST*	SETUPS RING  A DPTR, #CHARMEM SFILL LKB STPSEO A STPSEI A, ERRORS STPSE ERROR  line #8 table contains the jump tabl	; IF any char THEN exit ; Ring bel]  ; REPEAT point to char memory ; fill with char ; look at keyboard ; IF any char THEN exit ; bump to next char ; UNTIL all combinations ; UNTIL errors ; Jump to error routine  e for line #8 in setup mode.

1 1677	32				DB	/2/	·
1 1A77 2 1A78	32 19ED			•••••	DB	'2' 'STP8B	; Fill screen
3 4 1A7A	33				DB	/37	
5 1A7B	1A21				DW	STP8C	; Chanse attributes
6 7 1A7D 8 1A7E	34 1A3A				DB DW		
8 1A7E	1A3A 0004			L8TL	DW EQU	STP8E (\$-L8T)/3	; Continuous test ; Table length
9 10	%. Υ.Υ.Υ						
11 12							······································
13 14				<b>,</b> ,	L8T2 -	line #8 table 2	
15							table for line #8 in fill screen
16 17				; ;	*L812* option		table for line #0 in fire screen
18 19	1A80			L8T2	EQU	\$	
20							
21.,1A8Q 22.1A81	\$C				DW DW	KB-SPACE STP8B3	; Access code (ignore)
23							
24 25.1883	QD				DB	CR	
26 1A84	19DD				DW	SETUP8	; Carriage return
27 28 1A86	88				DB	KB_SETUP	. College and and
29. 1887 30	1AF.6				DW	SETUPRET	; Exit setup mode
31.1687	Ç8				ДВ DW	KB_SETUP+.KB_SHF SETUPSAVE	T; Exit setup mode and save
32 1A8A 33	1B02						
34 35	0004			L8TL2	EQU	(\$-L8T2)/3	; Table length
36							
37 38							
39 40					L8.=.).	ine.#8	
41					*L8*.d	ise)ays.setue.info	mation for line #8.
42 43				; 			
44 45				; ;	ENTRY	none	
46				,	EXIT	none	
47 48				<del>,</del>	USES	all	
.49							
50 511A8C.	E4			L8:	ÇLŖ	Α	; Position
52 53.1A8D.	7.1	90			AÇALL.	XŞT	
54 1A8F	2A	9D	45		DB	*MENU G* 1.	CHAR SET', (' '+80H)
1A92. 1A95	4E	55 2A	20 20				

729POM•	MICPOR	ENCH O		ASSEMBLER											. <b></b>
***Z-2	2.COMPU	TERTER	MINAL.F.I	ASSEMBLER RMWARE.VERS.	(VI)-218 [QN.1.Q3	24-FEB-8 ***	33 <b>15:30:</b> (	06 PAGE 2	:08+ 						
				20 41											
	1A9E .1AA1	43 52	48 20	41 53									•••••••		
	1AA4	45	54	53 Ao		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •					
							• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •		••••••			
***************************************	••••••		• • • • • • • • • • • • • • • • • • • •	•••••••	• • • • • • • • • • • • • • • • • • • •				••••••	• • • • • • • • • • • • • • • • • • • •					
************		• • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	•••••	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •			
***************************************	• • • • • • • • • • • • • • • • • • • •		•••••	· · · · · · · · · · · · · · · · · · ·	• • • • • • • • • • • • • • • • • • • •				•••••	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •				
		• • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		
***************************************			• • • • • • • • • • • • • • • • • • • •						• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		
•••••		• • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			•••••								• • • • • • • • • • • • • • • • • • • •	
•••••	• • • • • • • • • • • • • • • • • • • •				• • • • • • • • • • • • • • • • • • • •								• • • • • • • • • • • • • • • • • • • •		
		• • • • • • • • • •													
••••••	•••••														
••••••					•••••								· · · · · · · · · · · · · · · · · · ·		
			• • • • • • • • • • • • • • • • • • • •												
									• • • • • • • • • • • • • • • • • • • •						
•••••							• • • • • • • • • • • • • • • • • • • •								
												••••••			
												• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		••••••
											•••••••				• • • • • • • • • • • • • • • • • • • •
							•••••						•••••		• • • • • • • • • • • • • • • • • • • •
***************************************	• • • • • • • • • • • •			•••••								• • • • • • • • • • • • • • • • • • • •	••••••		• • • • • • • • • • • • • • • • • • • •
*************		• • • • • • • • • •	•••••••	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •		•••••	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		
***********		• • • • • • • • • • • • • • • • • • • •		•••••			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •			•••••		·····
***************		•••••	• • • • • • • • • • • • • • • • • • • •	••••••					•••••			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
*************		•••••		•••••	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •					• • • • • • • • • • • • • • • • • • • •			
•••••		• • • • • • • •					• • • • • • • • • • • • • • • • • • • •								

<u>1</u>									
····-5·									
	1.AA7	90	1AE4			MOV	DPTR, #LN8B1	; Primary	
3	1,444	30 90		03		JNB MOV	ACHR2, L8A		
4	1AAD	90	1AED			MOV	DPTR, #LN8B2	; Auxiliary	
Ś	1ABO	71	A8		L8A:	ACALL	XSTAT	; Character	set
···· <del></del>									
	1AB2	71	90			ACALL	XST		
····ģ	1AB4	<u>71</u>	9D 20	20	• • • • • • • • • • • • • • • • • • • •	ACALL.	2. FILL SCRE	N 3. ATTRIBUTES	<i>y</i>
	1AB7	20	22	25		22			
	1ABA	20 20	32 46	2E49					
		20	40	20					
	1ABD	4C	4C	<u>50</u>					• • • • • • • • • • • • • • • • • • • •
	1AC0	03 45	43	3Z					
	1AC3	53 45 20	45 20	4E 20					
	1AC6	20	20	20					
	1AC9	20 20	33	<u>2</u> E					
	1ACC	20	41	54					
	1ACF	54	52 55	49					
	1AD2	42	55	54					
	1AD5	<b>45</b> 20	53 20						
9	1AD7	20	20	20		DB	4. TEST', (	7+80H)	
	1 ADA	20	34	2E					
	1ADD	20 20	34 54 54	2E					
	1AEO	53	54	A0					
10									
11	1AE3	22				RET			
12	1AE3								
. 13.									
14									
. 15.									
16					; ;	Line #	8 setup mode messas	 25	
17									
18									
19	1AE4	ΑF	6F	72	LN8B1:	DB	'normal ', (1 1+)	BOH)	
	1AE7	6E	61	72 60					
	1AEA	20	20	A0					
	T.L.	20 61	20 	7 <u>.</u>	LN8B2:	DB	'alternat', ('e'+	30H)	
	1 / 10 / 10		00	4 =	LIVODZ.	DD	arternat 7 C c		
20	1AED	45							
20	1AFO	65	/2						
20	1AED 1AFO 1AF3	65 61	72 74	A0 74 6E					
20	1AFO	65 61	72 74	E5					
20	1AFO	65 61	74	E5					
20	1AFO	65 61	74	E5					
20	1AFO	65 61	74	E5					
20	1AFO	65 61	74	É5					
20	1AFO	65 61	74	E5					
20	1AFO	65 61	74	E5					
20	1AFO	65	74	E5					
20	1AFO	65 61	74	E5					
20	1AFO	65 61	74	E5					
20	1AFO	65	74	E5					
20	1AFO	65 61	74	ES					
20	1AFO	65	74	ES					
20	1AFO	65	74	E5					
20	1AFO	65	74	E5					
20	1AFO	65	74	ES					
20	1AFO	65	74	ES					
20	1AFO	65	74	ES					
20	1AFO	65	74	ES					
20	1AFO	65	74	ES					

Z29R ***	OM; Z-29.	MICROBE .COMPUT	NCH 805 ER.JERM	1 CROSS A INAL FIRM	SSEMBLER WARE YERS	(V1)-218 24-I IQN 1.03 ***	FEB-83 15	:30:06 PAGE 210	)			•••••••	
	1					<del>3.</del> <del>3</del>	SETUPRI	ET.H.netuno.fno	om setup mode				
	2 3 <b>4</b>					5 5 7		RET* exits setu lues that have		ut permaneni	lx.saxina		
•••••	5 6 7			• • • • • • • • • • • • • • • • • • • •		; ;		none	······································			• • • • • • • • • • • • • • • • • • • •	
	8 9 10					; ;		попе				•••••	
	10 .11 12					; ;		none					
	12 13	1AF6									***************	• • • • • • • • • • • • • • • • • • • •	
	15	1AF6 1AF9 1AFC	30 20 02	37 15 1E69	03 06	SETUPRET:	JNB JB JMP	L250N, SRET L25EN, SRET DISPSTAT	# 	IF 25th lir IF not enat display s	e is on ANI led for use	) r THEN	
	17	1AFF 1B00	.E4 .80	2C		SRET:	CLR SUMP	A SCLRL		ELSE clear			
•••••		• • • • • • • • • •	•••••	• • • • • • • • • • • • • • • • • • • •					•••••	• • • • • • • • • • • • • • • • • • • •		***********	
•••••	• • • • • •		•••••							• • • • • • • • • • • • • • • • • • • •	•••••••••	• • • • • • • • • • • • • • • • • • • •	•••••
•••••											•••••	• • • • • • • • • • • • • • • • • • • •	
•••••			• • • • • • • • • • • • • • • • • • • •					• • • • • • • • • • • • • • • • • • • •		•••••		• • • • • • • • • • • • • • • • • • • •	
••••••	• • • • • •		• • • • • • • • • • • • • • • • • • • •					•••••			• • • • • • • • • • • • • • • • • • • •		
•••••••						• • • • • • • • • • • • • • • • • • • •							
••••••	• • • • • •			• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •						• • • • • • • • • • • • • • • • • • • •	•••••
•••••		• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	•••••							• • • • • • • • • • • • • • • • • • • •	
• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •										•••••	
•••••	• • • • • •		• • • • • • • • • • • • • • • • • • • •										
•••••			•••••	• • • • • • • • • • • • • • • • • • • •				••••••				• • • • • • • • • • • • • • • • • • • •	
••••••			• • • • • • • • • • • • • • • • • • • •							• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	
		• • • • • • • • •		• • • • • • • • • • • • • • • • • • • •									
• • • • • • • • • • • • • • • • • • • •	• • • • • •			• • • • • • • • • • • • • • • • • • • •								• • • • • • • • • • • • • • • • • • • •	
				• • • • • • • • • • • • • • • • • • • •								• • • • • • • • • • • • • • • • • • • •	
		•••••										• • • • • • • • • • • • • • • • • • • •	
								• • • • • • • • • • • • • • • • • • • •				• • • • • • • • • • • • • • • • • • • •	
***************************************			• • • • • • • • • • • • • • • • • • • •							• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	
												• • • • • • • • • • • • • • • • • • • •	
			• • • • • • • • • • • • • • • • • • • •								• • • • • • • • • • • • • • • • • • • •		
								• • • • • • • • • • • • • • • • • • • •					

					SETUPS	AVE – save setup conf	figuration
2 3					*SETUP	SAVE* returns all val	lues back to the EAROM.
5				7  8	ENTRY	none	
<del>7</del>					EXIT	none	
9 10	. <b></b> .				ÜSES	all	
11				,		e. i i	
12 13 180 14 180	2F	1	.22 3 <b>4</b>	SETUPSAVE:	ACALL	NVRCHK	; Checksum the NVRAM ; Update checksum value
15					MOV	NVRSUM, A	
16 1BO	56 7	3			MOV MOV	PTR1, #25H DPTR, #EAROM	; EAROM store
17 1B0: 18 1B0:	B 7	ý A	7000 10		MOV	INDX1, #16	; 16 bytes
19 20 1B0	D E			SAV2:	MOV	A, @PTR1	; REPEAT
21 1809 22 1800	)EC.	4			SWAP MOVX	A @DPTR, A	; store upper nibble
23.1 <b>8</b> 1	.o A	3			INC	DPTR	
24 1B1 25 1B1	.1 C	4 )			SWAP MOVX	A @DPTR, A	; store lower nibble
26 1B1	3 A	3			INC	DPTR	
27 181 28 181	.40 5 D	3 A	 F6		INC DUNZ	PTR1 INDX1, SAV2	; bump pointers ; UNTIL done
29 30 1B1		 2			CLR	EAROMSAVE	
31.1 <b>8</b> 1	.9P.	1	.18		ACALL.	WLATCH	; Save EAROM
32 1B1 33 1B1			00 .18		SETB ACALL	EAROMSAVE WLATCH	; By togaling flag
34 35.1 <b>9</b> 1	.F 4	1	.F.4		AJMP	SETUPRET	; Return from setup mode
							······································

) ) ) ) ) )

•

Z29ROM; MICROBENCH 8051 CROSS ASSEME ***.Z-29.COMPUTER.TERMINAL.FIRMWARE	BLER (V1)-218 2 VERSION 1.03 **	24-FEB-83 15 **	:30:06 PAGE 212		
				•••••	
34	5 	*SFILL	* fills the complete s	creen with the data given it	
5		25th.1	ine however.		
	;	ENTRY	(A) = character to u	 Prite	
9			(DPTR) = memory loca	ition to start at	
11	;				
13 14 15			UP.1R11NUX1/2		
16 1821 7A 18 17 1823 7B 50	SFILL:	MOV MOV	INDX1, #MAXLINE INDX2, #NUMCHARS	; Go through all lines ; REPEAT	•••
18 1B25 F0	SFÎL2:	MOVX	@DPTR, A DPTR	; REPEAT ; REPEAT write characte ; bump pointer	r
20 1827 DB FC 21 1829 DA F8		DJNZ	INDX2, SFIL2	; UNTIL line is done ; UNTIL screen is done	
22 1828 22		RET			
					;
		• • • • • • • • • • • • • • • • • • • •	•••••		
			,		
	•••••		••••••		
		• • • • • • • • • • • • • • • • • • • •			••• •••••••••
				•••••	••••
					•••••
	•••••	• • • • • • • • • • • • • • • • • • • •			••••
				•••••	
	· · · · · · · · · · · · · · · · · · ·				
	•••••	• • • • • • • • • • • • • • • • • • • •		······································	

C C C C C C C C

	1 2			<del>5</del> <del>5</del>	SCLRLR SCLRL	— setup clear line re — setup clear line	verse video
	3 4 -				**************************************	R* clears 25th line wi	th reverse video spaces.
	5 6 7 8			,		* clears 25th line and on that line.	Places the attributes passed
13					·····ENTRY	···(A) = lower four bit	s used for attributes
	2				EXIT	(A) = unchansed	
1: 1: 1:	4				บระร	all	
1	7 1B2C	74	01	SCLRLR:	MOV	A, #0000001B	; Set for reverse video
	8 9 1B2E 0 1B30			SCLRL: i8	PUSH MÓV	ACC B, #MAXLINE CLRLINE	; Save attributes ; Clear line before display
2	1 1833 2 1836	<u>12</u>	0B96 E0		CALL	ACC	; Restore attributes
2	3 4 1838 5 1838 6 1836	・・・・ラク・・・・ 53 ・・・フA	5780 83 (	EF	MÖV ANL MÖV	DPTR, #L25MEM DPH, #ATRANL INDX1, #NUMCHARS	; Point to attribute memory
2	7 1B40 8 1B41	F0 A3		SLP:	MOVX INC	@DPTR, A DPTR	; REPEAT write attribute ; bump RAM pointer
	9 1B42 0 1B44	DA 22	FC		DJNZ RÉT	INDX1, SLP	; UNTIL done

				::	GETNUI	M.m.get.nymber.string.	
		• • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	**************************************			
		• • • • • • • • • • • • • • • • • • • •			*GETŅ	UM*.gets.a.number.from	.9255 from the keyboard. The
	<del>.</del>			, <del>,</del>	"DELE	"KEIUKN" finishes the TE" clears the complet	number and "BACKSPACE" or s.number.entered. The fourth
				;	chara	cter received automati	cally makes the routine return.
			• • • • • • • • • • • • • • • • • • • •	***************************************		•••••	
					ENTRY	(A) = Position on 2	5th line to start
					EXIT	(A) = number receiv (WORK2) = number of	ed (0255) characters picked up
		• • • • • • • • • • • • • • • • • • • •			USES	all	
				· · · · · · · · · · · · · · · · · · ·			
1B45	FE			GETNUM:	MOV	WORK1, A	; Original position
1B46	7F	00		GNUMO:	MOV		
1848	7F 7C	<u>00</u>	• • • • • • • • • • • • • • • • • • • •	CINCIPO	MOV	WORK2, #0 WORK, #0	; Position count ; Running total
					Get c	naracter	
1B4A	12	1407		; GNUM1:			······ <u>·</u> ···
10411				GNUM1:	CALL	SFCKB	; Get next character
				<b>5</b>	Check	for end of line	
1B4D	B4	OD	02		CUNE	A, #CR, GNUM2	; IF carriage return THEN
1B50 1B51	EC 22		• • • • • • • • • • • • • • • • • • • •	GNUMR:	MÖV RET	A, WORK	i get number and return
					Chank	for backspace or dele	T:
1852 1855	84 80	03 03	02	GNÚM2:	CUNE SUMP	A, #BS, GNUMZA GNUMZB	; IF backspace OR
1857	B4	7È	ÖĖ	GNUM2A:	CUNE	A, #RUBOUT, GNUM4	; IF delete THEN
			• • • • • • • • • • • • • • • • • • • •	·····;	Clear	current number	
185A	EF			; GNUM2B:	MOV	A, WORK2	
1B5B	60	ED	• • • • • • • • • • • •		JZ	GNUM1	; IF no chars THEN skip it
1850 185E	1E DF	FD		GNUM2L:	DEC	WORK1	REPEAT decrement position
	,		• • • • • • • • • • • • • • • • • • • •	••••••	DJNZ	WORK2, GNUM2L	; UNTIL count = 0
1860 1861	EE	9D	• • • • • • • • • • • • •		MOV	A, WORK1	; get position
1B63	71 20	9D 20	20		ACALL DB	XST	; wipe out old number
1866	Āŏ		<del></del>	•••••		/ /, (/ /+80H)	
1867	so	gg	• • • • • • • • • • • • • • • • • • • •		SJMP	GNUMO	; start over again
							. Start over asalii
				, ,		number and position	
1B69	BF	03	02	GNUM4:	CUNE	WORK2, #3, GNUM4A	; IF all chars THEN
186C 186E	80 B4	E2		Chillian A.A.	SUMP	GNUMR	; exit
180E 1871	50	3A D7	00	GNUM4A: GNUM4B:	CUNE UNC	A, #'9'+1, GNUM4B GNUM1	; Check upper bounds

C

(

4	1070	ъл	20	00		C. INE	A. #YOY. GNUMAC	
<u>1</u> .	1873 1876	<u>84</u>	30 b2	00	GNÚM4C:	CJNE Jč	A, #'O', GNUM4C GNUM1	; Check lower bounds
3							ate next number and	dieplay
5					, ,	Carcon	ate next number and	ulariai
	1878	94	30			SUBB	A, #/0/	; Subtract ASCII offset
	1B7A	CO	ΕO			PUSH	ACC	; Save it
	1B7C	cc				XCH	A, WORK	
	1B7D	75 A4	FO	OA		MOV	B, #10	
10	1880	A4				MUL	AB	; num := num * 10
	1881	2C FC				ADD	A, WORK WORK, A	
	1882	FC				MOV		; num := num + work
	1B83	BO	E0 1893			POP MÓV	ACC ''DPTR, #GNUMS''''	; Restore number
	1885	90	1893			ADD	A, DPL	
	1888 1888	25 F5	82 82			MÖV	DPL, A	; Bump address (caution)
	188A 188C	EE	82			MOV	A, WORK1	Get Position
	.188D	<u>55</u>	A8			ACALL	XSTAT	; Display
	188F	OF.	MO			INC	WORK1	Bump Position
	1890	OE				INC	WORK2	\$ Bump count
	1B91	80	B7			SUMP	GNUM1	; UNTIL done
22				• • • • • • • • • • • • • • • • • • • •				
23					;	Must n	ot cross any page b	oundaries
·····žă					• • • • • • • • • • • • • • • • • • • •			
	1B93	во	B1	B2	GNUMS:	DB	(101+80H), (111+8	OH), (′2′+80H), (′3′+80H), (′4′+80H)
	1B96	B0	B1 B4					
			T. /	VV				
26	1898	85	86	B7		DB	(15.480H) (16.48	OH), (′7′+80H), (′8′+80H), (′9′+80H)
27 28		B5 B8	B6 B9			DB IFC	NE,HIGH(\$)-HIGH(G	iNUMS)
27	1B9B 	B8	B9	¤/				iNUMS)
27 28	1B9B 	B8	B9			IFC ERROR	NE,HIGH(\$)-HIGH(G	iNUMS)
27 28	1B9B 	B8	89 89			IFC ERROR	NE,HIGH(\$)-HIGH(G	iNUMS)
27 28	1B9B 	B8	86 89			IFC ERROR	NE,HIGH(\$)-HIGH(G	iNUMS)
27 28	1B9B 	B8	B6 B9			IFC ERROR	NE,HIGH(\$)-HIGH(G	iNUMS)
27 28	1B9B 	BS BS	89 89			IFC ERROR	NE,HIGH(\$)-HIGH(G	iNUMS)
27 28	1B9B 	B8	B6 B9			IFC ERROR	NE,HIGH(\$)-HIGH(G	iNUMS)
27 28	1B9B 	B8 B5	B9			IFC ERROR	NE,HIGH(\$)-HIGH(G	iNUMS)
27 28	1B9B 	B8	B6 B9			IFC ERROR	NE,HIGH(\$)-HIGH(G	iNUMS)
27 28	1B9B 	B8 B9	B6 B9			IFC ERROR	NE,HIGH(\$)-HIGH(G	iNUMS)
27 28	1B9B 	B5 B8	B9			IFC ERROR	NE,HIGH(\$)-HIGH(G	iNUMS)
27 28	1B9B 	B8 B8	B6 B9			IFC ERROR	NE,HIGH(\$)-HIGH(G	iNUMS)
27 28	1B9B 	B5 B8	B9			IFC ERROR	NE,HIGH(\$)-HIGH(G	NUMS)
27 28	1B9B 	B8 B8	B6 B9			IFC ERROR	NE,HIGH(\$)-HIGH(G	iNUMS)

)

	<u>.1</u>		• • • • • • • • • • • • • • • • • • • •		XST -	transmit string	to 25th status line	
	2 3			; ;			of displayable characters that	
	4 5			<del></del>	follow	the CALL onto t	he 25th status line of the screen.	
			*****************	; :	ENTRY	(A) = position	on OEAh line	
	7 8				ENIRT	String to be s	ent must immediately follow the call	
i			•••••		•••••	seven set.	e. Final character must have bit	
<u>1</u>	<u>1</u> 2	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	<del>,</del>	EXIT	(A) = next pos	ition on 25th line	
	3 4				USES	A, DPTR, PTR1		
	5 &							
1	7 1B9D 8 1B9F	po 	83 82	XST:	POP POP	DPH DPL	; Get return address	
	0 1BA1							
2:	1		A8		ACALL	XSTAT	; Output string	
2:	2 1BA3 3 1BA5	ço	82 83		PUSH PUSH	DPL DPH	; Place return address ; back onto the stack	
2	4 1BA7	<u>00</u>			RET	· · · Ŧ / · · · · · · · · · · · · · · ·		
······						• • • • • • • • • • • • • • • • • • • •		

	1				YSTAT -	- transmit strine to 25t	h status line
	<u></u>	· · · · · · · · · · · · · · · · · · ·					
	3 4 5				*XSTATe	f places a string of dis th status line of the sc	Playable characters onto reen.
				· · · · · · · · · · · · · · · · · · ·	ENTRY	(A) = position on 25th	line
	7 8 9			; ; ;	ENITY	(DPTR) = address of st	ring, final character must have
. ,	10 11 12			<del>,</del>	EXIT	(A) = next position or (DPTR) = position foll	25th line owina strina
	13 14 15			; ;	USES	A, DPTR, PTR1	
	16 17 1BA8	75	A0 57	XSTAT:	MOV	P2, #HIGH (L25MEM)	; Set address of memory
	18 1BAB	75 24 F8	A0 57 80		ADD MOV	A, #LOW (L25MEM) PTR1, A	; Add in offset to low byte ; Point to memory
	20						, , , , , , , , , , , , , , , , , , ,
	.21 1BAE 22 1BAF	<b>E4</b> 93		XSTL:	CLR	AA, @A+DPTR	; Get data byte
	.23 .1BBQ	A3			INC	PRTR	; Bump ROM pointer
	24 1BB1	20	E7 04	ļ.	JB MOVX	ACC.7, XSTR @PTR1, A	; IF bit set THEN exit out ; Output data
	.25 .1BB4 26 .1BB5	F.2 08		• • • • • • • • • • • • • • • • • • • •	INC	PTR1	; Bump RAM pointer
	.271BB6	80	F.6		SJMP	XSTL	; Go forever
	28 29 1BB8	54	7F	XSTR:	ANL	A, #7FH	; Mask off high bit
	30 1BBA	F2	//		MOVX	@PTR1, A	; Output last data
	.31.1BBB		81		MQY		; Get pointer
	32 1BBC 33 1BBE	2 <b>4</b> 22	81		ADD RET	A, #-LOW (L25MEM) + 1	; Subtract offset ; Return
		<del>6/4</del>	••••••				
		· · · · · · · · · · · ·					

) ) ) ) ) ) ) )

1					; ;	XLONOFF	- tra	nsmit strine "	'ON" or "OFF"	to 25th status	s line	
2 3					; :					icated by the d		
4					*	flas.	A COLOMBIA	(.,,4.,4,80,80		A 9.94 9.2. 9. 9. 9. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.	591.4.7	
9 6			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	;		• • • • • • • • • • • • • • • • • • • •					
<u>7</u>						ENTRY	= (A) = (c)	Position on 2 O indicates "	25th line "OFF" - 1 ind	icates "ON"		
9 10						EXIT		next position				
${12}^{10}$					, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,				. ON 23th 11n	e		
13					<b>,</b>	USES	A, DP	TR, PTR1		•		
14	1BBF	90	1809		XLONOFF:	MOV	пото	#LNOFF				
16	"1802"	····5ö	03		XLONOIT.	JNč	XLON1		; IF can	he "OFF" messas rry set THEN	a	• • • • • •
	1BC4 1BC7	<u>90</u>	1BCC A8		XLONT:	MOV	DPTR,	#LNON	set	the "ON" messa ay message and	ase	
<u>19</u> 20			H0		XLUNI:	AJMP	XSTAT		; Displa	ay message and	return	
21												
·····żż						· · · · · · · · · · · · · · · · · · ·	• • • • • • • • • • • • • • • • • • • •				• • • • • • • • • • • • • • • • • • • •	
23 24	1BC9 1BCC	6F	66 6E	<u>E6</u>	LNOFF:	DB DB	(of/.)	(′f′+80H) (′ ′+80H)				
		• • • • • • • • • • • • • • • • • • • •										
				• • • • • • • • • • • • • • • • • • • •								

				SEMBLER (V1)-218 2 NARE VERSION 1.03 **		:30:06 PAGE 219	
	1				XMTS -	transmit strins	
_	2 3 4 · · · · · · · · · · · · · · · · · · ·			; ; ;	*XMTS*	transmits a string less of whether the	of characters out of the terminal terminal is on line or not.
	6 7 8 9			\$ \$ \$ \$	ENTRY	String to be sent to this routine. seven set.	must immediately follow the call Final character must have bit
1:	1			;	EXIT	none	
12 13	3			; ;	USES	A, DPTR	
15		DO DO	83 82	xmts:	P0P P0P	DPH DPL	; (DPTR) = first character
18	8 9 1BD3	E4		VMTC1 =	CLR		
20 2: 2:	0 18D4 1 18D5 2 18D8	93 20 12	E7	XMTS1: 06	MÖVC JB CALL INC	A A, @A+DPTR ACC.7, XMTS2 TRANSMIT DPTR	; REPEAT fetch character ; IF bit 7 set THEN exit ; transmit character ; bump pointer
24	3 1BDB 4 1BDC	8ò	F5		SJMP	XMTS1	; UNTIL forever
25 20 21	5 6 1806 7 1861	<u>.</u> 12	139B 01	XMTS2:	CALL MOV	TRANSMIT A, #1	; Transmit last character
	8 1BE3	<b>74</b> 73			JMP	@A+DPTR	; Return to code after string
		•••••					
•••••							
*****************			• • • • • • • • • • • • • • • • • • • •				
· · · · · · · · · · · · · · · · · · ·			•••••				
v							
************							

					VMT:	toppemit changeter	
: 2		• • • • • • • • • • • • • • • • • • • •		? ?		transmit character	
<u> </u>							ter at the current cursor Position
<del> </del> 5				; •			escape sequences for entering and
'		• • • • • • • • • • • • • • • • • • • •					tribute modes for the mode the racter is protected then no
7 						is taken.	······································
3				;			
, )		• • • • • • • • • • • • • • • • • • • •			ENTRY	(A) = current trans	smit mode
<u>.</u>							to screen coordinate
 2				;	EXIT	$(\Delta) = current trans$	smit mode (updated)
}				· · · · · <del>[</del> · · · · · · · · · · · · · · · · · · ·			smit mode (dradted)
5 				<b>,</b>	USES	all	
, 7							
1BE4	çö	21	• • • • • • • • • • • • • • • • • • • •	XMTC:	·····PUSH···	ATTRIBUTES	; Save attribute byte
1BE6	co	E0			PUSH	ACC	; Save current attributes
1BE8	30	25	OA		JNB	GATM, XLO	; IF not GATM THEN
IBEB	12				CALL	CPF	; check for protected
1BEE	50	05			JNC	XLO	; IF protected THEN
71860″ 5 18F2	DO DO	E0			POP	ACC	; restore values
1BF4	22	21			······································	ATTRIBUTES	; and return
, 3 1865 -							
	E5	11	F0	XLO:	MOV	A, XPOS	
) 1BF7 ) 1BFA	<u>85</u>	12 0503	F0		MOV CALL	B, YPOS CLA	; Calculate line address
							, carcarate rine address
2718FD77 3	Do	Fo			POP	В	; Get current attributes
TBFF	53	83	EF		ANL	DPH, #ATRANL	; Mask to attribute memory
1002	EO				MOVX	A, @DPTR	
LCQ3	····F5····	21	4.0		MOV	ATTRIBUTES, A	; Get current attributes
1005 1008	43 Eo	83	10		ORL MOVX	DPH, #CHRORL A, @DPTR	; Mask to character memory
1009	A2	E7			MOV	C, ACC.7	: Get alt char flas
TCOB	92	ōć			MQV	ALTCHARA, C	; Place in proper location
100D 100F	62	7F öb			ANL	A, #7FH	; Mask alternate char set
: ICOF	UZ.	OΠ			ĊĽŔ	GRPH	; Clear graphic flag
tctt	B4	50	<u>o</u> o		CJNE	'''A, #''''', XLÒA	
1014 1016	40 184	7 <del>5</del>	öĖ	XLOA:	JC JC	XL1	; IF graphic char THEN jump
1016	D4	/٢	UF.		CUNE	A, #RUBOUT, XL3	; IF displayable THEN jump
				· · · · · · · · · · · · · · · · · · ·	Charac	ter is graphic	
) Stranska series				<b>;</b>			
1019 101B	74 80	5E			MÖV	Α, # ^{2λ2}	; Map 7Fh to 5Eh
		X			SJMP	XL2	
1C1D	B4	1F	04	XL1:	CUNE	A, #1FH, XL1A	; Special case map
1020	74				MOV		
1022 1024	80	02 		XL1A:	SJMP ORL	XL2 A, #01100000B	; Make graphic printable
	• •	~~		ri lui di ET ■	OIL	H, #01100000B	) Have aldellic Lithianie

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (VI)-218 24-FEB-83 15:30:06 PAGE 221 *** Z-29 COMPUTER TERMINAL FIRMWARE VERSION 1.03 *** Save character to output PUSH See if character is graphic character ; IF not graphic AND GRPH, XL4 ; IF previous was graphic THEN 10 11 1C2D 5D JNB F5. B.5 12 1030 . C2... CLR: 13 IF ANSI mode THEN 30. 40 ANSIT XLBC 14 1032 IF previous not alt THEN JB B.4, XL3B 15 1035 ÖC UB ... ALTCHARA; XL3A IF this not alt THEN 12, LC38, 20 71 ACALL XMTS transmit "EXIT GRAPHIC MODE" 17 1C3B 58 DB .... 18 103D 1B 1040 31 19 1042 49 ACALL XMTS XI 3A: 21 1044 22 1046 1B ESC, 1(1, (11/+80H transmit "ENTER ALT MODE 23 1049 80 XL4SET flas in alt now 24 ELSE IF this not alt THEN ALTCHARA; XL3A 25 1048 20 transmit "ENTER NORMAL" 71 ACALL 26 1C4E ESC, 2(27, (282480A) 28 ĎΒ 27 1050 18 CLR B.4 XL7 28 1053 C2 flas in normal now SUMP 29 TC55 30 .....xL3C: XMTS ACALL 31 1057 ESC, ('G'+80H DB 32 1059 1B C7 XĽŸ 33 TC5B .80 30 34 ijġ..... ; IF previous not graphic THE 35 TC5D B.5 D2 F5 SETB flas chanses 36 1060 ANSI, XL4C B.4, XL4B 38 1062 30 20 IF ANSI mode THEN F4 ÜĖ. IF previous normal THEN 39 1065 IF not alt char THEN ALTCHARA, XL4A oc JB 40 1068 20 7i ĊĖ XMTS transmit "ENTER GRAPHIC XL4AA: ACALL 41 1C6B 42 1C6D ESC, [[10], ([m]+80H) 1B 5B DB 1070 30 ËĎ 43 1072 80 19 45 1C74 71 1B CF ACALL transmit "ENTER ALT GRAPHIC · 28 46 1C76 DB ESC. B.4 flas in alt now 47 1079 D2 XL4SET: SETB 48 1C7B .80 10 49 JB 20 öċ. 50 1C7D XL4B: ALTCHARA, XL4A ELSE this is normal THEN XMTS transmit "ENTER NORMAL" 51 1080 71 CF ACALL .... 28 ESC, 52 1082 18 DB B.4 CLR 53 1085 flas in normal now.... 54 1087 80

1 1089	71 1B	CF		XL4C:	ACALL	XMTS	; ELSE transmit in ZDS mode
2 108B	1B	C6			DB	ESC, (/F/+80H)	
4				**************************************	See if	character is in reve	erse video
.5 6 108D		D5	• • • • • • • • • • • • • • • • • • • •	; 	CLR	F0	; Clear "CHANGE" flas
7 8 108F		os	OE		JB		; IF current not rev video AND
8 108F 9 1092	20 30		0E 1B		JB JNB	RVVA, XL8 B.O, XL9	; IF current not rev video AND ; IF previous was rev video THEN
0 1095	30 C2	F0 F0		• • • • • • • • • • • • • • • • • • • •	CLR	B.0	; clear rev video
1 211097	26	4ö	i4		JB	ANSI, XL8A	; IF ZDS mode THÉN
3 1C9A	71	CF			ACALL	XMTS	; output "EXIT REV VIDEO"
4 1090 5 109E	1B	F1			DB	ESC, (/q/+80H)	
9 ICAE	80	10			SJMP	XL9	
7 1CA0	20	FO	OD	XL8:	JB	B.O, XL9	; IF previous not rev video THEN
8116A311 9	р́2	Fö	• • • • • • • • • • • • • • • • • • • •		SETB	B.0	; set rev video
oricas	····żö····	40	99		JB	MANSI, XLBA	; IF ZDS mode THÉN
1 1CA8 2 1CAA	71	CF Fö			ACALL	XMTS	; output "ENTER REV VIDEO"
2 10AA 3 10AC	1B 80	02			DB SJ <b>M</b> P	ESC, (YPY+80H) XL9	
4							
5 1CAE	D2	D5		XL8A:	SETB	F0	; ELSE flas chanses
7				;	See if	character is underli	ine
ś				; ;			
9 1CBO	20	09	07	XL9:	JB	UNDLNA, XL10	; IF not underline AND
1 1CB3	30	F1	OB		JNB	B.1, XL11	; IF previous was underline THEN
2 1CB6	C2	F1			CLR	B.1	; clear underline flag
3 1CB8 4	80	05			SJMP	XL10A	; flas chanses
5 1CBA	20	F1	04	XL10:	JB	B.1, XL11	; IF previous not underline
91GBD	DZ	F1	• • • • • • • • • • • • •		SETB	В.1	; flas chanses
7 1CBF	D2	D5		XL10A:	SETB	F0	
9				;	See if	character is blinkin	19
1 1CC1	20	OA	07	; XL11:		BLINKA, XL12	
2					JB		; IF not blinking AND
3 1004	30	F2	OB		JNB	B.2, XL13	; IF previous was blinking THEN
V 1007 V 5 1009	80	F2			CLR SJMP	B.2 XL12A	; clear blinking flag
5 1009		05			SUMP	XLIZA	; flag changes
7 1CCB 3 1CCE	<u>20</u>	F2 F2	04	XL12:	JB	B.2, XL13 B.2	; IF previous not blinking THEN
9 1CDO	D2 D2	F2 D5		XL12A:	SÉTB SETB	B.2 F0	; flas chanses
3			• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	
l 2				<del>5</del> 5	See if	character is half in	itensity
3 1CD2 4	20	OB	07	XL13:	JB	HALFIA, XL14	; IF not half AND
5 1CD5	<u>30</u>	F3 F3	ов		JNB	B.3, XL15	; IF previous was half THEN
icos —	C2	FЗ			CLR	B.3, XL15 B.3	; clear half flas

				ERSION 1.03 *			
1CDC	20	F3	64	XL14:	Jġ	B.3, XL15	; IF previous not half THEN
1CDF	.20 D2		04	VC14.	SETB	B.3	; flag changes
1CE1	D2 D2	F3 		XL14A:	SETB	FÖ	
					See if	character is altern	ate graphic
							· · · · · · · · · · · · · · · · · · ·
10E3	20	oc	19	XL15:	JВ	ALTCHARA, XL16	; IF not alt AND
1CE6	30	F4	31		JNB	B.4, XL17	; IF previous was alt THEN
1CE9	C2	F4			CLR	B.4	; clear alt flas
1CEB	30	40	2A		JNB	ANSI, XL16B	; IF ANSI mode THEN
1CEE	71	ĊF			ACALL	XMTS	; transmit "EXIT EVERYTHING"
1CFO	1B	28	C2		DB	ESC, ((', ('B'+80H	)
1CF3	30	F5	24		JNB	B.5, XL17	; IF graphics mode THEN
1CF6	71	ĊF			ACALL	XMTS	; transmit "ENTER GRAPHIC MODE"
1CF8	1B	5B ED	31		DB	ESC, /[10/, (/m/+80	0H)
1CFB	30				C MD	VI 47	
1CFD	80	1B			SJMP	XL17	
.10FF	20	F4	18	XL16:	J.B.	B.4, XL17	; IF previous not alt THEN
1D02	D2	F4			SETB	B.4	; flaa chanaes
1D04	30	40	11		JNB	ANSI, XL16B	; IF ANSI mode THEN
1D07	20	F5 	07		JB	B.5, XL16A	; IF no graphics THEN
1DOA	71	CF			ACALL	XMTS	; transmit "ENTER ALT MODE"
1DOC	1 B	28	B1		DB	ESC, ((1, ((1/+80H	)
1DOF	80	09			SJMP	XL17	
1D11	71	CF	• • • • • • • • • • • • • • • • • • • •	XL16A:	ACALL	XMTS	; ELSE transmit "ENTER ALT GRAPHIC
1D13	1.B	28	B0		DB	ESC, 1(1, (101+80H.	)
1D16	80	02			SJMP	XL17	
1D18	D2	D5		XL16B:	SETB	FO	
				·····;	Transm	nit any chanses	
1D1A	30	p5	46	XL17:	JNB	F0, XL20	; IF no changes THEN skip
					JB	ANSI, XL18	; Check for mode
1D1D	20	40	1F				, check for mode
				;	ZDS mo	ode chanses	
1D20	71	CF			ACALL	XMTS	; set attributes
.1022	1.B	F.3			DB	E\$C, (/s/+80H)	
1D24	CO	FO			PUSH	В	; save attributes
1D26	E5	F.Q			MQV	A. B	; reorder bits
1D28	13				RRC	A	; get reverse video
.1029	13				RRC	<u>A</u>	; get underline ; put in position
1D2A	92	F3			MOV	B.3, C	
1D2C	13				RRC	A	; get blink ; put in position
1D2D 1D2F	92 13	F1			RRC	A	get half intensity
	43 92	F2			MOV	B.2, C	
1D30 1D32	E5				MOV	A, B	

		(Eù.''; Eù;	итичг ЕТК	MWARE. VI	ER\$ION.1,03*;	*.*		
1	1D34	5 <b>4</b>	1F			ANL	A, #00011111B	; mask proper bits
2	1D36	54 24	30	· · · · · · · · · · · · · · ·	• • • • • • • • • • • • • • • • • • • •	ADD	A, #'0'	; add offset for sequence
	1D38	12	139B			CALL		; output attribute data
4	1D3B	DO	FO			POP	В	; restore attributes
	1.D3D	80	24			SJMP	XL20	
6	•				_			
 8			• • • • • • • • • • • • • • • • • • • •			ANSI.A	oode chanaes	
		71	CE		, XL18:	ACALL	XMTS	; clear any prev attributes
16	1D3F 1D41	71 1B	CF5B	BO	AFA9*	ACALL.	^!!!! ESC, /[/, (/o/+80H)	; crear any prev accributes
11			O.D	20		22	250, 1, 10, 100117	
	1044	30	F3	04		JNB	B.3, XL18B	; IF half intensity THEN
	1047	71 3B	CF B2			ACALL	XMTS	
	1049	3B	B2			DB	XMTS 7;7, (727+80H)	
	1D4B	30	F1	<u></u>				
		30	F1	04	XL18B:	JNB	B.1, XL18C	; IF underline THEN
	1D4E	71 3B	CF B4			ACALL.	XMTS 7;7, (747+80H)	
19		3D	D4			DB	';', ('4'+80H)	
	1D52	30	F2	04	XL18C:	JNB	B.2, XL18D	; IF blinking THEN
	1D55	71	ÇF	~ ,	NEI CO	ACALL	XMTS	7 II DIAMATIS IOLIA
22	1D57	3B	B5	• • • • • • • • • •		DB	/;/, (/5/+80H)	
23								
	1D59	30	FO	04	XL18D:	JNB	B.O, XL18E	; IF reverse video THEN
	1D5C	71	CF			ACALL	XMTS /;/, (/7/+80H)	.,
	1D5E	зв	B7			DB	(;/, (/7/+80H)	
27	1060	71	CF					
28	1D60	71	CF		XL18E:	ACALL	XMTS	; output last character
	1D62	ED				DB	(/m/+80H)	
		DO	EΩ		XL20:	POP	ACC	
32	1D63 1D65	<u>DO</u>	E0 139B		AFAY:	POP CALL	TRANSMIT	; Output character
33							113111321121	
34	1D68	<u>63</u>	Fo	3F		ANL	B, #00111111B	; Mask previous
35	1D6B	E5	21 3F			MOV	A, ATTRIBUTES	
	1D6D	54	ЗF			ANL	A, #00111111B	; Mask current
37	1D6F	<u>45</u>	FO			ORL	A, B	; Combine to make new previous
	1071	DO	2 <b>i</b>			POP	ATTRIBUTES	; Restore true attributes
39	1D73	22				RET		
	• • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •		
	• • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	••••••
						• • • • • • • • • • • • • • • • • • • •		
			• • • • • • • • • • • • • • • • • • • •					
	• • • • • • • • • •			• • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			•••••••••••••••••••••••••••••••••••••••

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLE *** Z-29 COMPUTER TERMINAL FIRMWARE VE	RSION 1.03 ***				
1	;;	XMTL -	transmit line entry to transmit line		
2	<b>*</b>				
4 5	;	*XMTL* by tran	transmits the entire cont smit character *XMTC*.	ents of the line as defined	
6	,	*XMTL1*	needs a current transmit	mode.	
7					
9 10 11	· · · · · · · · · · · · · · · · · · ·	ÉNTRÝ	(A) = current transmit m YPOS points to screen li	ode (*XMTL1*) ne	
11 12 13 14	;	EXIT	(A) = current transmit m	ode (updated)	
14	<del>,</del>	USES	all		
15 16		7			
17 18 1074 E4	XMTL:	····clr	A	; Start with no attributes	
19 1D75 CO 11	XMTL1:	PUSH MÖV	XPOS XPOS, #O	; Save current position ; Start at beginning of line	
21 1D7A 71 E4	XML2:	ACALL	XMTC	REPEAT transmit character	
22 1D7C 05 11 23 1D7E AC 11		INC MOV	WORK, XPOS	; bump position	
24 1D80 BC 50 F7		CUNE	WORK, #NUMCHARS, XML2 XPOS	; UNTIL done : Restore position	
25 1D83 D0 11 26 1D85 FC		····MOV	WORK, A	; Restore position ; Save (A)	
27 1D86 12 04EA 28 1D89 EC		CALL	BLINAD A, WORK	; Update address ; Restore (Å)	
29 1D8A 22		RET		; Return	

**)** ) )

)

•

)

)

)

Z29ROM; MICR *** Z-29 COM	OBENCH 80	51 CROSS ASSE	EMBLER (V1)-218 2 RE VERSION 1.03 **	4-FEB-83 15	:30:06 PAGE 226	
					- special transmit	character
3 4 5	• • • • • • • • • • • • • • • • • • • •		; ; ;	*SXMTC: Positi	* transmits the cha on as defined by tr	racter at the current cursor ansmit character *XMTC*.
6  8			7 7	ENTRY	none	
8 9 10				EXIT	none	
10 11 12			; 	USES	all	
13 14 1588	ვგ	40 OF	SXMTC:	JNB	ANSI, SXC1	
15 1D8E		09D1		CALL JZ	GNP XMTP	; IF ANSI mode THEN ; get parameter
17 1D93 18 1D94	14	12		DEC UZ	A SXMTL	; IF zero THEN xmt page ; IF one THEN xmt line
19 1D96 20 1D97	14	ö4		DEC	A SXC1	; IF two THEN xmt character
21 1D99 22 1D9A	14	2A		DEC	A XMT25	; IF three THEN xmt 25th line
23 1D9C 24		• • • • • • • • • • • • • • • • • • • •		RET		; ELSE return
25 1D9D 26 1D9F		3F	SXC1:	PUSH MÖV	PFIELD PFIELD, #0	5 Save protected fields 3 Make none protected
27 1DA2 28 1DA3	71	É4		CLR ACALL	A XMTC	; No attributes set ; Transmit character
29 1DA5 30 1DA7		3F		POP	PFIELD	; Restore protected fields
00 LD(()	22			RET		/ Nestore Protected Fields
	22					
	22					
	22					
	22					
	22					
	22					
				RET		
				RET		
				RET		
				RET		

C C C C C C C C C

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLEF *** Z-29 COMPUTER TERMINAL FIRMWARE VEF	R (V1)-218 24-FE RSION 1.03 ***	B-83 15:	30:06 PAGE 227		
	5 5	SXMTL -	- special transmit line		
1 2 3 4 5	; ; ;	*SXMTL*		ontents of the line as defined The end of the line is signaled	
6 7 8 9 10		···EŃTŔY···	none		
10	<del>7</del>	EXIT	none		
11 12 13		ÚSÉS	all		
13 14 15 1DAS B1 74 16 1DAA 80 14	SXMTL:	ACALL SUMP	XMTL XMP3	; Transmit line ; Send CR, bell, and return	

Z29R0N *** Z-	M; MICROE	BENCH 805	51 CROSS ASSEMBI	_ER (V1)-218 /ERSION 1.03.*	24-FEB-83 15 **	:30:06 PAGE 228		
	.1					transmit page		
	2 .3			; ;	*XMTP*	transmits the en	tire contents of the page	· (excluding
	4		• • • • • • • • • • • • • • • • • • • •	,			tire contents of the page y transmit character *XMT	
	6		• • • • • • • • • • • • • • • • • • • •		QTINE	Page.lsslgnaled	.by.a.canriage_return	••••••
	.7 8	• • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	· · · · · ·	ENTRY	none		
	.9 10	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		EXIT	none		
	11		• • • • • • • • • • • • • • • • • • • •		USES			••••
	12 13			<del>,</del> 	USES	all		
	14 151DAG	E4		XMTP:	CLR	Α	; Start with no	attributes set
1	16 1DAD	E4	12		PUSH	YPOS	; Save current r	
1	171DAF 18 1DB2	75 B1	1200 75	XMP1:	MOV ACALL	YPOS, #0 XMTL1	; REPEAT transmi	t line
	19.1DB4 20 1DB6	05 AC	12		INÇ	YPOS WORK, YPOS		
	21.1DB8	BÇ	İ\$F.7		CJŅE	WORK, #MAXLINE,	XMP1 ; UNTIL done	
	22 23.1DBB	po	12	XMP2:	POP	YP.OS	; Restore positi	on
2	24 1DBD	12	04EA		CALL	BLINAD	; Update address	
2 2	26 1DCO 271DC2	71 8D	CF	XMP3:	ACALL DB	XMTS (CR+80H)	; Send CR to ter	minate page
2	28 29. 1003	02	066B		JMP	RING	; Signal done to	. USAP
		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •				•••••	•••••
	• • • • • • • • • • • • • • • • • • • •		••••••		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		
			• • • • • • • • • • • • • • • • • • • •					
							•••••	
					•••••••	• • • • • • • • • • • • • • • • • • • •	•••••	
		• • • • • • • • • • • • • • • • • • • •	•••••••		• • • • • • • • • • • • • • • • • • • •	•••••	••••••	
• • • • • • • • • • • •			•••••	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		
,								
							• • • • • • • • • • • • • • • • • • • •	
					••••••	•••••••		
	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		•••••	
	• • • • • • • • • • • • • • • • • • • •	•••••		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		•••••	

	RÓM; Z-29	MICROE COMPL	ENCH 805 TER TERM	1 CROSS	ASSEMBLE RMWARE VE	R (V1)-218 2 RSION 1.03 **	4-FEB-83 15: *	:30:06 PAGE 229			
	1					<b>;</b> ;	XMT25 -	- transmit 25th lin	e		
	2 3 4 5					; ; ; ;	*XMT25		h line in is enable	the same manner as *XMTL* ed. The end of the line	
							·····ÉNTRÝ	none			
	9 9						·····EXIT····	none			
	11 12 13					<del>5</del>	······································	ali			
	14	 1DC6	30	15	F7	XMT25:	JNB	L25EN, XMP3	• · · · · · · · · · · · · · · · · · · ·	IF 25th line enabled THEN	
	17	1DCB 1DC9	30 C0 75 B1	15 12 12	18		MOV	YPOS, #MAXLINE		save current position do 25th line	
	18	1DDO 1DCE	B1 80	12 74 E9			ACALL SJMP	XMTL XMP2		transmit 25th line restore, update, return	
			• • • • • • • • • • • • • • • • • • • •								
•••••											
•••••									**************		
• • • • • • • • • • • • • • • • • • • •											
•••••											
*********											
• • • • • • • • • • • • • • • • • • • •											

)

.1		• • • • • • • • • • • • • • • • • • • •		PRINT	— print page	
4 5		•••••	; ; ;	the da	* acts exactly like * ta goes out to the or iage return and line	XMTP* (transmit page) except that  posite port and every line ends with feed.
6 7			; ;			
9			# #	ENTRY	none	
.9 [0	• • • • • • • • • • • • •	•••••	*	EXIT	none	
. <u>1</u> .2 .3				USES	all	
4 5 1DD2	D2	23	PRINT:	SETB	PRNTF	; Flas printins
6 7 1DD4	12	1304	•••••••••	LCALL	DOXOFF	
8						; Output an XOFF
9 1DD7 0	D1	OF	•••••	ACALL	CMPORT	; Togale port
1 1DD9 2 1DDB 3	C0 75	3F 3F		PUSH MOV	PFIELD PFIELD, #0	; Save protected fields ; Make all unprotected
4 IDDE	E4			CLR	А	; Start with no attributes set
5 1DDF 6 1DE1	<del>75</del>	<u>12</u>	 00	PUSH MÖV	YPOS YPOS, #O	; Save current position
7 8 10É4	B1	<del>75</del>	PRNT1:	ACALL	XMTL1	; REPEAT transmit line
9 1DE6	CO	E0		PUSH	ACC	; save attributes
O IDES 1 IDEA	71 OD	88		ACALL DB	XMTS CR, (LF+80H)	transmit CR and LF
2"1DEC" 3	ро	EQ		POP	ACC	; restore attributes
4''1DEE'' 5 1DFO	05' AC	12		inc	YPOS	; bump position
6 TDF2"	BC	12 18	EF	ÖÜNE	WORK, YPOS WORK, #MAXLINE, PRN	T1 ; UNTIL done
7 811116511	po	12			YPOS	
9 1DF7	12	04EA		LCALL	BLINAD	; Restore position ; Update address
1 1DFA	DO	3F		POP	PFIELD	; Restore protected fields
2 3 1DFC	D1	OF		ACALL	CMPORT	; Tossle port asain
1 5 1DFE	20	22	03	JB	HSSTOPF, PRNT2	
5''1E01'''	12	ໂຮ້໓ຮ…		ECALL	DOXON	; IF not hold screen THEN ; output an -XON-
3 1E04	c2	23	PRNT2:	CLR	PRNTF	; Flag done printing
91 <b>E</b> 02	12	02E8		LCALL	LKB	; Look at keyboard
l 1E09	50	03		JNC	PRNT3	; IF no keyboard data THEN
3 1EOB 1 1EOE	02 22	066B	PRNT3:	LJMP RET	RING	; signal done to user ; ELSE return
						· · · · · · · · · · · · · · · · · · ·

Z29 ***	ROM; MICR Z-29 COM	DBÉNCH 80 PUTER TER	51 CRÓSS MINAL FI	ASSEMBLE RMWARE VE	R (V1)-218 2 RSION 1.03 **	4-FEB-83 15 *	:30:06 PAGE 231		
	1						- complement port		
	3 4				7 <del>7</del>	*CMPOR	T* toggles direction	of the I/O port.	
•••••	<u>5</u>				<del>.</del>	ENTRY	none		
•••••	7s				,	EXIT	none		
• • • • • • • • • • • • • • • • • • • •	…າ່ຽ 11					······U\$E\$	A, DPTR		
	12 13 1EOF 14 1E12 15 1E14 16 1E16	B2	13 02 30 16	FD	CMPORT:	JNB CPL CPL CLR	OKTRANS, CMPORT PORTF PRTF2 XOFFRCVED	; Wait until tran ; Toggle port fla ; Toggle EAROM fl ; Clear any recei	<b>3</b>
	17 18 19				· · · · · · · · · · · · · · · · · · ·	SJMP	WLATCH	; Write and retur	n
	19 20 21								
	21 22 23 24				<u> </u>	WLATCH	— write latch		
	<u>25</u>				; ;	-WLATC	H- writes the curren memory mapped latch	t value of the byte $*XL$	ATCH* out
	27 28 29 30				* * * * * * * * * * * * * * * * * * *	ENTRY	none		
	30 3 <b>1</b> 32				; ;	EXIT	none		
	32 33 34				; 	USES	A, DPTR		
	35 36 1E18 37 1E1A 38 1E1D	90 F0	20 3000		WLATCH:	MOV MOV MOVX	A, XLATCH DPTR, #MLATCH @DPTR, A	; Get byte ; Get address ; Output data to	latch
	391E1E	22				RET			
• • • • • • • • • • • • • • • • • • • •									
•••••									
• · · · · · · · · · · · · · · · · · · ·									

2 3 4 5					DCLK -	- Display clock	
4 5				;		- always display clo	- k
5		• • • • • • • • • • • • • • • • • • • •		***************************************			
6		• • • • • • • • • • • • • • • • • • • •			*DCLK	* displays the current once every second only	t setting of the clock on the 25th
7							
o 9				, <del>.</del>		O* always displays the 5th line.	e current settins of the clock on
o 1				;			
2 3			• • • • • • • • • • • •	; ;	ENTOV		
3 4	• • • • • • • • • • • • • • • • • • • •				ENTRY	none	
5 6				<del>.</del>	EXIT	none	
7 8		• • • • • • • • • • • • • • • • • • • •		, , , , , , , , , , , , , , , , , , ,	USES	A, B, DPTR	
9	<u></u>						
0 1E1F 1 1E21	E5 B5	3E 3B	01	DCLK:	MÓV CJNE	A, DISPSEC A, TSEC, DCLKO	. IE no shape in sounds TUEN
2 1É24	22				RÉT	H) TOECY BEEKO	; IF no change in seconds THEN ; exit
3 4 1625	₈₅	зв	ЭĒ	DCLKO:	MÖV	DISPSEC, TSEC	; Init *DISPSEC*
5 1E28	E5	3D			MOV	A, THOUR	
6 1E2A	12	0A92			CALL	CBA	; Convert hours to ASCII
7 1E2D	<u>69</u>	82 83			PUSH	DPL	; Save ones digit
8 1E2F	E5				MOV	A, DPH	; Get tens digit
7 1E31	B4	30	92		CUNE	A, #101, DCLK1	; IF tens digit = 'O' THEN
0 1E34	74	20			MOV	Α, #	; suppress leading digit
1 1E36 2 1E39	90 F0	57C8		DCLK1:	MOV	DPTR, #L25MEM+72	
		<b>50</b>			MOVX	edetr, A	; Place on screen
3 1E3A 4 1E3C	B3	E0			POP	ACC	
					INC	DPTR	
5 1E3D	<del>7</del> 4	3A			MOVX	@DPTR, A	; Place on screen
21E3E		3A			MOV		
7 1E40 3 1E41					INC	DPTR	
9 1E41		00			MÓVX	edetr, A	; Place ':' on screen
) 1E44	E5 12	3C 0A92			MOV	A. TMIN	
1E44	CO				CALL	CBA	; Convert minutes to ASCII
1E47		82 83			PUSH	DPL	; Save ones disit
	E5				MOV	A, DPH	
1E4B	F0	57CB			MOV	DPTR, #L25MEM+75	***************************************
		<b>F</b> 0			MOVX	edetr, A	; Place tens digit on screen
1E4F	D0 	E0			POP	ACC	
1E51					INC	DPTR	
'1E52 '1E53	<del>F</del> 0	з <u>а</u>			MOVX	@DPTR, A	; Place ones disit on screen
		JH.			MOV	A, #*:	
) 1E55 ) 1E56	A3				INC	DPTR	
		on.			MÓVX	edetr, A	; Place ':' on screen
1E57 1E59	E5	3B			MOV	A, TSEC	
		0A92			CALL	CBA	; Convert seconds to ASCII
3 1E5C	<u>co</u>	82			PUSH	DPL	; Save ones disit
1E5E	E5	83			MOV	A, DPH	
1E60	20	57CE			MOV	DPTR, #L25MEM+78	
1E63 1E64	FO DO	ΕO			MOVX POP	@DPTR, A ACC	; Place tens disit on screen

**(**)

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FE *** Z-29 COMPUTER TERMINAL FIRMWARE VERSION 1.03 ***	B-83 15:	30:06 PAGE 233	
1 1E66 A3 2 1E67 F0 3 1E68 22	INC MÖVX RET	DPTR @DPTR, A	; Place ones digit on screen
	· · · · · · · · · · · · · · · · · · ·		
	••••••	•••••	
•••••••••••••••••••••••••••••••••••••••			
•			
••••••			
***************************************			

					(V1)-218 24-F ION 1,03 ***	EB-83 15	:30:06 PAGE 234			, • •
	1				j.j	DISPST	AT - display status	line		
	2 3 4 5  7				5 5 7 5	the 25: slack:	TAT* displays every th line if the opti caps locked, on/of ter mode.	on is enabled.		
	3				; ;	ENTRY	none			
					5 5 3	EXIT	none			
	3				<del>,</del>	USES	all			
17 i	5 1E69 7 1E6C 8 1E6F	30 30 22	37 15	01	DISPSTAT: DSPR:	JNB JNB RET	L250N, DSPR L25EN, DSPO	; IF 1:	ine 25 off OR ine 25 enabled THEN turn	
	9 0 1E70 1 1E72	7 <b>4</b> 71	08 2E		DSPO:	MÓV ACALL	A, #00001000B SCLRL	; Clear	with half intensity	
	2 3 1E74 4	D1	25			ACALL	DCLKO	; Disp	lay clock	
2! 2: 2: 2: 2:	5 1E76 5 1E77 7 1E7A B 1E7D	E4 90 30 71	1E94 35 A8	02		CLR MÓV JNB ACALL	A DPTR, #MSGCL CAPLOCKF, DSP1 XSTAT	; Caps	locked	
3: 3:	9 0 1E7F 1 1E81 2 1E84 3 1E87	74 90 20 71	10 1EA5 2E A8		DSP1:	MÓV MOV JB ACALL	A, #16 DPTR, #MSGOL ONLINE, DSP2 XSTAT	; Off1:	ine	
::::::::::::::::::::::::::::::::::::::	5 1E89 5 1E8B 7 1E8E	74 90 30	1D 1EB3 21 A8	02	DSP2:	MOV MOV JNB	A, #29 DPTR, #MSGIC ICMODEF, DSP3		rt character	••••
39	5 1E91 9 5 1E93	71 22	A8 		DSP3:	·····ACALL	XSTAT	; Oute		
•••••			•••••							
			• • • • • • • • • • • • • • • • • • • •							
1979 6			• • • • • • • • • • • • • • • • • • • •	••••						• • •
*************		•••••	• • • • • • • • • • • • • • • • • • • •							
E 717 7/2 21					*****					
****************										•••

Z29ROM; MYCRÓBENCH 8051 CRÓSS ASSEMBLI *** Z-29 COMPUTER TERMINAL FIRMWARE VI	ER (V1)-218 24-F ERSION 1.03 ***	EB-83 15:30:06 PAGE 235
1	;	Caps locked message
3 1E94 00 20 20 1E97 43 41 50	MSGCL:	DB OOH, / CAPS LOCKED /, (OOH+80H)
1E9A 53 20 4C 1E9D 4F 43 4B		
4 5 6		Off line message
6 7 8 1FA5 00 20 20	, MSGOL:	DB OOH, / OFF LINE /, (OOH+80H)
8 1EA5 00 20 20 1EA8 4F 46 46 1EAB 20 4C 49 1EAE 4E 45 20		200 2112 7 100110011
1EAB 20 4C 49 1EAE 4E 45 20 1EB1 20 80		
10		
10 11 12	<del>,</del>	Insert character message
13 1EB3 00 20 20	; MSGIC:	DB OOH, ' INSERT MODE ', (OOH+80H)
1EBC 20 4D 4F		
1EBF 44 45 20 1EC2 20 80		
••••••		
		••••••
		•••••••••••••••••••••••••••••••••••••••
		•••••••••••••••••••••••••••••••••••••••
	•	
	•••••	
	•••••	

ļ		• • • • • • • • • • • • • • • • • • • •		КВСНК	- keyboard check	
2			;	*KBCH!	<pre>&lt;* waite for the keyboan</pre>	d to send -KB_POWER- within
		• • • • • • • • • • • • • • • • • • • •	**************************************	75 ms.	. If it is not found th	en an error condition exists.
j	· · · · · · · · · · · · · · ·			This.d	routine must anly be don	e.after a hardware reset so that is soins to send the data.
, , 			·····	10 15	KHOWH CHAI CHE KEYDOAFG	is soins to send the data.
} >			;	FNTRV	none	•
)			***************************************			
				EXIT	none	
ŀ			<b>,</b>	USES	all	
; ; .						
1EC4		····od	KBCHK:	MOV	INDX1, #13	
1EC6 1EC8	7B i 2	00 02B6	KBCO: KBC1:	MOV CALL	INDX2, #0 FCKB	; REPEAT init count ; REPEAT set any data
1ECB	40	04 8F		JC	KBC2	; REPEAL SET ANY DATA ; IF data AND
1ECD 1EDO	B4	8F	F8	CUNE RET	A, #KB_POWER, KBC1	; IF power up THEN ; return
1ED1	22 DB	F5	KBC2:	DUNZ	INDX2, KBC1	; UNTIL done
1ED3	DA	F1		DJNZ	INDX1, KBCO	; UNTIL all done
	43	40	08	ORL	ERRORS, #00001000B	; ELSE error #3
5 1ED5 5 1ED8	43 22	40	08	ORL RÉT	ERRORS, #00001000B	; ELSE error #3
	43 22	40	08	ORL RÉT	ERRORS, #00001000B	; ELSE error #3
	43	40	08	ORL ŘÉŤ	ERRORS, #00001000B	; ELSE error #3
	43 22		08	ORL RÉT	ERRORS, #00001000B	; ELSE error #3
	43 22		08	ORL RÉT	ERRORS, #00001000B	; ELSE error #3
TEDS				RET		; ELSE error #3
TEDS				RET		
TEDS				RET		

Z29ROM; *** Z-2	MICROBENCH 805	11 CROSS ASSEMBI IINAL FIRMWARE '	_ER (V1)-218 24 VERSION 1.03 ***				
1			<b>5</b> 5		- ROM checksum		
3	! }		; ,	-ROMCHI	K- does a checksum on th	e ROM and compares it with	• • • • • •
5			.,	the kn	own value in KUM.		
	) •			ENTRY	none		• • • • • • • • • • • • • • • • • • • •
	) 			EXIT	none		• • • • • • •
10 11			; 	USES	all		
12 13	}		DOMOUNA.	MÓV	WORK, #0		
15	1ED9 7C 5 1EDB 90 5 1EDE 7A	00 0000 20	ROMCHK:	MOV MOV	DPTR, #0 INDX1, #20H	; Init checksum ; Init pointer ; First index	
1.7	1EEO 7B	00		MOV	INDX2, #00H	; Init count	
18 19	1EE2 E4		RMC1:	CLR MOVC	A A, @A+DPTR	; REPEAT ; set ROM byte	
21	) 1EE3 93 1EE42C 2 1EE5 FC			ADD MOV	A, WORK WORK, A	; add to previous amount ; replace	
	3.1EE3	F9		INC DJNZ	DPTR INDX2, RMC1	; bump pointer ; UNTIL done	
	5.1EE9DA	Fź		DUNZ	INDX1, RMC1	; UNTIL all done	
	, 1EEB 40 1EED 43	03 40 01		JZ ORL	RMC2 ERRORS, #00000001B	; IF error THEN ; error #0	
29	2.1EF022		DMCO.				
	+ m'. X <del>/. //</del>	•••••	RMC2:	RET		•••••	
			RPCZ:	RET			
			RNICZ:	RET	.,		
			RMCZ:	RET			
			RMC2:	RET			
			RNL2:	RET			
			RNICZ:	RET			
			RNICZ:	RET			
			RNICZ:	RET			
			RNICZ:	RET			
			RNICZ:	RET			
			RNICZ:	RET			
			RNICZ:	RET			
			NIICZ:	RET			
			RNICZ:	RET			

1					RAMCHK	( - RAM check	
2				;	FIAMCI	UZ	Cailuana It united two
 4		• • • • • • • • • • • • • • • • • • • •			differ	HK- checks the RAM for any cent values into character	and attribute RAM and reads
.5						ack. Any errors are flas	
6				<b>5</b>			
./ 8		• • • • • • • • • • • • • • • • • • • •			ENTRY	none	
?							
O 1				, ,	EXIT	none	
[2]				5	USES	a'l	
13 14		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		
5 1EF1	90	5000		RAMCHK:	MOV	DPTR, #CHARMEM	; Point to character RAM
6 1EF4 7	7C	55			YOM	WORK, #01010101B	; First pattern
 8 1EF6	ŻA	08	• • • • • • • • • • • • • • • • • • • •	RAMO:	MOV	INDX1, #OSH	; First index
9 1EF8	7B	00			MOV	INDX2, #OOH	; Second index
20 21 1EFA	E.C.			RAM1:	MOV	A, WORK	; REPEAT
11.4558 22 1EFB	EÇ FO			KAMI:	MUY MOVX	@DPTR, A	; KEFEA! ; write Pattern
3 1EFC	. 53 	83	EF		ANL	DPH, #ATRANL	; move to attribute RAM
4 1EFF	FO				MOVX	@DPTR, A	; write mattern
5.1F00	E0 54	oF			MOVX	A, @DPTR	; read pattern
6 1F01	54 FD	OF.			ANL MOV	A, #OFH TEMP, A	; mask ; save
7.1F03 8 1F04	<u></u>	• • • • • • • • • • • • • • • • • • • •			MÖV	A, WORK	; set orisinal pattern
9 1F05	54	of 			ANL	A, #OFH	; mask it also
0 1F07	B5		1 4		CUNE	A, TEMPA, RAMERR	; IF error THEN exit
1.1F0A 2 1F0D	43 E0	83	10		ORL	DPH, #CHRORL	; move to character RAM again
2 1F0D 3 1F0E	85	04	OD		MOVX CJNE	A, @DPTR A, WORKA, RAMERR	<pre>; read pattern ; IF error THEN exit</pre>
4 1F11	ĀĞ		· · · · · · · · · · · · · · · · · · ·	• • • • • • • • • • • • • • • • • • • •	INC	DPTR	; bump pointer
5 1F12	DB	E6			DUNZ	INDX2, RAM1	; UNTIL done
6 1F14 7	DA	E4			DUNZ	INDX1, RAM1	; UNTIL all done
ź. 1F16	····BC	55	04		CJNE	WORK, #01010101B, RAM2	; IF pattern is original THEN
9 1F19	7C 80	AA D9			MOV	WORK, #10101010B	; change to second pattern
D'iFiB'' 1	80	D9			SJMP	RAMO	; and do it asain
2"1F1D"	22	• • • • • • • • • • • • • • • • • • • •		RAM2:	RET		; IF no error THEN return
3 4''1F1E'''	₄₃	4o		RAMERR:	ÖRL	ERRORS, #00000010B	; ELSE flas error #1 and return
5 1F21	22		-		RET	militaria, hadaaaaa	7 Sandardan I Francis Griff St. III a Garline I G. C. Garline

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER *** Z-29 COMPUTER TERMINAL FIRMWARE VERS	(V1)-218 24-FE SION 1.03 ***	B-83 15:	30:06 PAGE 239		
1	;;	NVRCHK	- non-volatile RAM chec	ksum	
	<del>,</del>	*NVRCHk	(* does a checksum on th	e NVRAM and returns the value.	
4	; ;				
5 6		ENTRY	none		
		EXIT	(A) = checksum value o	f NVRAM	
9 10	. <del>.</del>	UŚEŚ	A, WORK, INDX1, PTR1		
11					
13 1F22 E4 14 1F23 78 25	NVRCHK:	CLR	A PTR1, #25H	; Init for checksum	
15 1F25 7A OF		MOV	INDX1, #15		
17 1F27 26 18 1F28 08	NVR1:	ADD INC DUNZ	A, @PTR1 PTR1 INDX1, NVR1	; REPEAT add next value ; bump RAM pointer ; UNTIL done	
19 1F29 DA FC 20 1F2B 22		RET		, on it and	
			•••••		
					• • • • • • • • • • • • • • • • • • • •
					••••••••••
,					

1					ERROR	- diagnostic error	
2 3				;			sages for any errors that
4				**************************************	may ha	ve been flassed in the -	-ERRORS- byte. The status
.Ə 6			• • • • • • • • • • • • • • • • • • • •			s.displayed.if.there.are when finished and no err	≥ no errors. A bell is
Ž					314611		ors occured.
8 9				;	Format	of the error byte is as	s follows:
. ó					BIT	O = ROM checksum error	······································
. <u>1</u>						1 = RAM error 2 = CRTC error	
.3				7		3 = Keyboard error	
5				7		4 = Non-volatile RAM o	thecksum error
.7					CNTDV		
.e					ENTRY	none	
.9 :0					EXIT	none	
1				, , , , , , , , , , , , , , , , , , ,	USES	all	
(2 :3							
4 1F20	E5	40		ERROR:	MOV	A, ERRORS	
5 1F2E 6 1F30	70 Di	05 69			JNZ ACALL	ERRO DISPSTAT	; IF no errors THEN
7 1F32	02	066B			LJMP	RING	<pre>; display status ; ring bell and return</pre>
8							
9 1F35 0 1F36	E4	2É		ERRO:	CLR ACALL	A SCLRL	
1 1F38	D1	25 öf			ACALL	DCLKO	5 Clear line full intensity 5 Display clock
2 1F3A	рі	öĖ			ACALL	CMPORT	; Change ports
3 4 163611	85	4o	Fö		MOV	B, ERRORS	; Get errors
5		CF					
6 1F3F 7 1F41	71 45	CF 52	52	ERM1:	ACALL DB	XMTS 'ERROR -/, (/ /+80H)	; Output message
1F44	4F	52	····žõ···			ENTOR - 3 ( FOOT)	
1F47 ≅	2D	A0					
9 1F49	E4				CLR	А	
0 1F4A	80	1F41			MOV	DPTR, #ERM1	•••••••••••••••••••••••••••••••••••••••
1 1F4D 2	71	A8			ACALL	XSTAT	; Output to 25th line
3 1F4F	30	FO	17		JNB	B.O, ERR1	; IF ROM checksum error THEN
4 1F52'' 5 1F53	71	CF			MOV	WORK2, A	
0 1655 6 1655	52	4F	4°D	ERM2:	ACALL	XMTS 'ROM checksum', (' '+	; output message 80H)
1F58	20	63	68	best 11 false "	22	NOT CHECKSOM 1 ( T	0917
1F5B	~~~65~~~	93	6B			• • • • • • • • • • • • • • • • • • • •	***************************************
1F5E 1F61	<u>73</u>	75 Ao	6D				
7 1F63	EF				MOV	A, WORK2	
8 1F64	90	1F55			MOV	DPTR, #ERM2	
9 1F67 3	71	A8			ACALL	XSTAT	; output to 25th line
0 1 1F69	30	F1	14	ERR1:	JNB	B.1, ERR2	; IF RAM error THEN

729ROM; MICROBENCH 8051 CROSS ASSEMBLER (VI) -218 24-FEB-83 15:30:06 PAGE 241 *** Z-29 COMPUTER TERMINAL FIRMWARE VERSION 1.03 *** 1 1F6C WORK2, A 2 1F6D 71... ACALL" XMTS. 'RAM fault 3 1F6F 52 20 1F72 75 1F75 1F78 20..... EF 4 1F7A A, WORK2 1F6F DPTR, #ERM3 5 1F7B 90..... MOV: 71 XSTAT 6 1F7E ACALL output to 25th line 8 1F80 30 JNB B.2, ERR3 ; IF CRTC error THEN MOV .... WORK27 A 9 TF83 FF CF 10 1F84 71 ACALL XMTS output messase 52 11 TF86 43 ERM4: //CRTCTennon1/,11(/11/480H) 1F89 43 ЪĖ. 1F80 72 72 20 1F8F 72 A0 12 1F92 EF. 'AJ WORKZ ... 1F86 13 1F93 90 MOV DPTR, #ERM4 14 1F96 71 ACALL ; output to 25th line B.37 ERR4 16 TF98 .30..... JNB IF keyboard error THEN ERR3: FF MOV WORK2, A 17 1F9B 71 18 1F90 ACALL 'XMTS' ; output message 19 1F9E 1FA1 62 72 1FA4 ÄÓ 1FA7 20 1FA8 EF A, WORK2 90 ĭF9Ė DPTR, #ERMS 21 1FA9 MÖV 22 1FAC 71 **ACALL** output to 25th line 23 24 1FAE IF NVRAM error THEN 30 JNB B.4, ERR5 FF MOV [25] [FB1] WORK2, A 71 ACALL XMTS 26 1FB2 56 ERM6: 27 1FB4 'NVRAM checksum', (' (H08+ 40 1FB7 41 .98 65 1FBA 63 75 1FBD 6B 6D 1FC0 28 1FC3 29 1FC4 EF A, WORK2 90 MOV 1FB4 DPTR, #ERM6 30 1FC7 71 A8 **ACALL** XSTAT output to 25th line 31 32 1FC9 **ACALL** XMTS ; Output CR, LF's ...od. 33 1FCB ĎB. CR, LF, CR, (LF+80H) BB.

		RAMCO	ONST - RAM constants		
	<b>;</b>	#RΔM0	CONST* is the table wi:	th all of the RAM cons	tants that
······································	• • • • • • • • • • • • • • • • • • •	the t	terminal will power up	with.	,
1FD1 DO	RAMCONST:	DB	LOW (IBUF)	; ISTORE	• • • • • • • • • • • • • • • • • • • •
1FD2 D0		<u>DB</u>	LOW (IBUF)	; IFETCH	
1FD3 00		DB	Ö	; ICOUNT	
1FD4 FO		DB	LOW (OBUF)	; OSTORE	
1FD5 FO		DB	LOW (OBUF)	; OFETCH	•••••
1FD6 00		DB	o	; XPOS	
1FD7 00 1FD8 50		DB DB	O HIGH (CHARMEM)	; YPOS ; LINADH	
1FD9 00		DB	LOW (CHARMEM)	; LINADL	***************************************
1FDA 00		ĎB	HIGH (NORM)	; DSPADH	
1FDB 87		DB	LOW (NORM)	; DSPADL	
1FDC 00		DB	0	; TSCROLL	
iFDD 18	• • • • • • • • • • • • • • • • • • • •	DB	O MAXLINĖ	; BFIX	***************************************
1FDE 00		<u>DB</u>	o	; PMADRH	
1FDF 00		DB	·····	; PMADRL	
1FE0 00 1FE1 00		DB	0 0	; FMNUM	
	• • • • • • • • • • • • • • • • • • • •	DB		; PMVALUE	••••••
1FE2 00 1FE3 01		DB		; KBIDNUM ; HSLINE	•••••
1FE4 00		DB	0	; ASLINE ; GSET	
1FE5 39		DB	00111001B		L:L: 44
		DB	ороророрв	ATTRIBUTES	bits #1 bits #2
1FE7 08 1FE8 0C		DB DB	00001000B	; Bits #3	
1FE8 OC 1FE9 30		DB	00001100B 00110000B	; Bits #4 ; Bits #5	
1FEA OO 1FEB OO	•••••	DB DB	o	; XXPOS ; XYPOS	
		DB	32		
1FED 00			32 0	; BLNKRATE ; BLNKCOUNT	
1FEE 00		DB DB	o	; AUTOCNT	•••••
1FEF00		bb	······o	; TĊNŤ	
1FF0 00		DB	ò	; TSEC	
1FF1 00 1FF2 00		DB DB	0 0	; TMIN ; THOUR	
1FF3 FF		DB	255	; DISPSEC	
1FF4 00		DB	0	; PFIELD	

Z29ROM; MICROBENCH 8051 CROSS A *** Z-29 COMPUTER TERMINAL FIRM	WARE VERSION 1.03 **	*					
	<del>,</del> 5	CHKS -	check sum				
4	, , , , , , , , , , , , , , , , , , ,	-CHKS- choose	is the checks n so that the o.	sum byte for the total sum of al	terminal ROM. I I the bytes sives	t is a value	
5 6 7 1FFF		ORG	1FFFH				
8					Shamadanan basaba		
9 1FFF D6	CHKS:	DB	од6Н	, , , , , , , , , , , , , , , , , , ,	hecksum byte		
11		END					
					••••	• • • • • • • • • • • • • • • • • • • •	
						•••••	
						•••••	
				• • • • • • • • • • • • • • • • • • • •			
						• • • • • • • • • • • • • • • • • • • •	
	,						
						• • • • • • • • • • • • • • • • • • • •	
						• • • • • • • • • • • • • • • • • • • •	
						• • • • • • • • • • • • • • • • • • • •	
······································						• • • • • • • • • • • • • • • • • • • •	
						• • • • • • • • • • • • • • • • • • • •	
•							
						•••••	
*****							

9Ç	%00D6	01	AÇA1	.0A96	. 03	ACA2		03	AÇA3		03	ACA3A	.0A12	03
ACA5	0A16	03		700E0	00	ACCESS	001E	01	ACD	0001	03	ACDN	OCBB	03
9CD1	0000	Q3	ACHR2	.003D	Q1	ACL	.QQ4Q	03	ACLET	QC46	93	ACLQ	QC4B	03
ACPR	OAAF	03	ACP1	OACE	03	ACP1A	OADS	03	ACP2	OADC	03	ACP3	OACC	03
9ÇR	9974	୧३	ASRI	∵óčěE…	9.3	AÇR1	.0073	03	ACU	0003	03	ACUP	. QÇ2D	03
ACU1	OCA2	03	ADA	OEC3	03	ADAM	OECC	03	ADM3	0042	01	ADM3ES	06A7	03
AESC1	06A1	0ु३	ALLINI	10AF		ALLI1	1001	03	ALTCHA	000C	01	ALTGRE	.000F	01
ANSI	0040	01	ANSIES	0697	03	ANSIT =		03	ANSITL=		00	APBT	0006	03
APBT1 APDL	OCCB	03 03	APCA APDLO	. 09FA	<u>03</u>	APDC	.0E32	03	APDCO	<u>0E37</u>	03	APDC1	.0E38	
APIC1	ODEF	03	APIL	ODBB ODZD	03 03	APDL1	ODBC	03	APIC	ODE9	03	APICO	ODEE	03
APRNT	ÖADF		ARAMP	0AF6		APILO ARM	0D82 085F	03 03	APIL1 =	OD83 0871	03	APR	OAE9	
ASBR	OAEA	03	ASBR1	OAEF	03	ASBR2	OAF2	03	ARMT = ASCP	OEEF	03 03		0005	00
ASGMT =			ASGMTL=			ASGO	OA4E	 03	ASGO	OA49	03	ASGM	OA3B	03  03
ASM	083D	03	ASMR	084F	03	ASMT =	0850	03	ASMTL =		00	ASG1 ASPF	0A65	03
ASPF1	0A68		ASPF1A	0A72		ASPF1B	OA77		ASPF1C			ASPF1D	0A81	
ASPF1E	0880	03	ASPF1F	0A8C	03	ASPF2	OASE	03	ATBC	062B	03	ATRANL=		<del> </del>
YTRIBM≐	4000		ATTRIB=			AT2	0635		AUSCP		ŏš	AUTOCN=		• • • • • • • • • • • • • • • • • • • •
AUTOCR		01	AUTOLF	002D	01	AUTOWR	002B	01	A1M	0880	03	A1RM	0800	03
41RT =	- 08DE	03	AIRTL =	0003	00	AISM	08B0	03	AISR	0802	03			03 03
11STL =		00	A2M	0898	03	A2RM	0944	03	A2SM	08FF	03		%00F0	00
BANKO =			BANK1 =	0008		BAUDRA=	.0033		BELL =	0007			0018	***************************************
BLINAD	O4EA	03	BLINKA	000A	01	BLINKF	0001	01	BLNKCN=	0038		BLNKRA=	0037	
BRTAB =		03	BR1	OFD2	03	BR2	OFDC	03		.0008		BTAB	0602	
BTL.	060A	03	BTL1	.061A	03	BTR	061F	. 03		0018		CAPLOC	0035	01
BA	0A92	03	CCRSR =			CDN	OCA8	03	CDN1	OCAE	03	CDN2	OCB2	03
DN3	OCB7	०ुटु	CHARME=			CHKS	1FFF	03	CHRORL=			CINT =		
LRFLG	0503 0020	03	CLFO	OBC2	03		OC3D	03	CLKF	0033	01	CLKRUN	0097	01
LRTB	0520	<u>01</u>	CLRL	OBAO 1EOF	03 03	CLRLIN	OB96	୍ର ପୁଞ୍ଚ	ÇLRS	OBAC	03	CLRTAB	.0598	
:PFR2	01A7	03	CPF1	018E	03	CONTLF CPF2	001D 01A3	01 03	CPF	0188	03	CPFR1	0180	03
PRO	OESB	<u>63</u>	CPROA	.018E		CPR1	OE97		CPR CPR2	.0E83 0EA6	03 03	CPRM	.0EA9	
RESET=			CRNBK	0038	01	CRT	0064	03	CRT1	OC6A		CR =	0000	01
Ri	1198	o.a	CR2	119A	ğŝ		.00E0		…蚆ôp∵≕		03	CSTRT ≡		
S1	OBB5	03	CS2	OBB9	03	CTAB1 =		03	CTAB1L=		00	CTAB2 =		03
TAB2L=	∷QQQQE	9 <u>0</u>	''CTAB3'≅		.оз	CTAB3L≅			cts	.00B5	Öi	CUA	∵ŏĊĎ6∵	
UA1	OCE4	03	CUA2	OCE8	03	CUA2A	OCEB	03	CUAS	OCEF	03	CUBO	OCF2	03
UB1	ÖCFE			ODO4			OD14		CUCOA	ÖDIC	93	CUCI	ODOE	
UC1A	OD12	03	CUC2	OD1E	оз	CUP	OCSD	03	CUP1	0094	03	CUP1A	0095	03
UPTB'''	0099	03	CUR	OCF1	.03	CURSOR	1184	03	CURI	ODOO		CUR2	ÖD20	03
	%00D7	01	DC	OB5F	03	DCLK	1E1F	.03	DCLKO	1E25	03	DCLK1	1E36	03
CMDRE≐ KI		00	DDATRE=			DEOL	OBOO	.03	DISPSE=			DISPST	1E69	03
KI MAORL≐	OFB9		DMAANL=		, . ,	DMADRH=			DMADRL=			DMAMEM=		
MPTRA=			DMATYP DM1	0084	01		40001		DMLNA =				70000	
MID	123D		DMIE	1225	03	DM1A DM2	1243	.03 .03	DM1B	1235	03	DM1C	1238	
OXOFF	1304	03	DOXON	13D5	03		1248 40083	03	DM3	124A		DM4	1258	03
SADAL=			DSPR	166F		··bspo····/	1E70	.03	DPL :	%0082 1E7F	00	DSADRH=		_.
SP3	1E93	03	DSR	OD24	03	DSRERR	OD40	03	DSROK	0D42		DSP2	1E89	
SROA	∵öbžč…	· 83	DSRi	OD30	ŏš	- DSRIA	.0035	.03	DSR2	.0D42	03 03	DSRO DSRŽA	OD29 OD3É	03 03
STARE=			DTR	0094	01	DX	0014	01	D25L	0B15	. 03		700AF	01
ACR ·····	089.5	. <u>93</u>	EALF	0B68	.93	··ĒĀM·····	0B03			7000		EAROMR	0003	
AROMS	0000	01	EBD	OBF1	03	EBL	OBCB	03	EBLNK	0F30	03	EB1	OBF8	03
B2	ÓBFA	.03	EB3	ocoi		··EC	085C	.93	EED	0008	ŏš	EEL	OBDB	
ER	0018	03	EERM	OB74	03	EE1	ococ	03	EE1A	OC10	03	EGATM	OB6E	. 03
GM	ÓF39	03			03	EGTTL =	0002	00	EHALF	OF2A		EHSM	овзс	
ICM	OB54	03	EID	OA1A	03	EID1	0A20	03	EID2	0A25	03	EIL	OA2B	03

C

IL1	0A31	03	EIL2	0A36	. 03	EIR	0A2A	03	EKAM	OB4E	03	EKC	OB26	03	•
KI	OFB3		EKSM	OB48	gg	ELB	07BC			0700		ELBTL =	001E	ÖÖ	
NBLCU	001A	01	EOL	OBEO	03	EOL1	OBE3	03	EQMT =	0892	03	EQMTL =	0002	00	
RM	001B	01	ERM1	1F41	03	ERM2	1F55	03	ERM3	1F6F	03	ERM4	1F86	03	
RM5	1F9E	03	ERM6	1FB4		ERROR	1F2C	03	ERRORS=			ERRO	1F35	<u>03</u>	 
RR1	1F69	03	ERR2	1F80	03	ERR3	1F98	03	ERR4	1FAE	03	ERR5	1FC9		
RVM To	0F33	03 01		%00AC %00AB	01 	ESC =	001B 0F2D		ESCCOD	067B %00A8	03 	ESCRET.	0696 20088	03 01	
ZM	%00A9 0B07	03	E11 E25L	OBOB	03	EUNDL FCIF	01F6	03	FCKB	02B6	03	FCOF	024C	03	
ĬĽĽ	1211		FII	OIFC	,	F12	0202		FIS	0217		FKDLY	02E1	оз	
KDLY1	02E4	03	FKL	0207	03	FKR	02DD	03	FK1	02BE	03	FK2	02D8	03	
Li	1219		FOR			F01	0253		FREQ	0039	Öi	FULLDP	002F		
	%00D5	01	GATM	0025	01	GDES	OF4F	03	GDESR	OF7A	03	GDESO	0F52	03	
ĎĔŚĩ	OF55		GDES2	OF5A	оз	GDES2A	0F62	.03	GDES3	OF65		GDES3A	0F71	öз	
ETNUM	1B45	03	GNP	09D1	03	GNPF	0019	.01	GNP1	09DC	03	GNP2	09E8	03	
NUMR'''	1850		GNUMS	1893	03	GNUMO	1B46	.03	GNUM1	1B4A	03	GNUM2	1852	03	
NUM2A	1B57	03	GNUM2B	1B5A	03	GNUM2L	1B5D	ల్లిక్రై	GNUM4	1B69	03	GNUM4A.	1B6E	03	
NUM4B``	1871	03	GNUM4C	1876 001F	03	GPICK	0F86	03	GPICKT≃	0F7B	03	GRPH	000D 0F47	01 03	
SEL IPICK	0012 OF81	 	GSET =		01	GODES	0F3F 0043	01	GOPICK HAZESC		03	G1DES	11D4	03	
	0768	03	HAZTL =		00	HCLFT	0051	03	HCLFT1	OC5F	03	HCRT	0079	03	
CRT1		3	····HEED			HEE1	· öczó	ŏš	HEE1A	0C24		HIR	iibs.	ŏ3	
NDSHK	003E	01	HSLINE=			HSMODE	0036	01	HSSTOP	0022	01	HT =	0009		
TAB	0636		HTC	0627	э	HTS	0621	03	HTSI	0623	03	нто	0640	оз	
T1	0652	03	HT2	065B	03	нтз	0665	03	IBUF =	57D0		ICMODE	0021	01	
COUNT	000E		ici	114F	03	IDT	OEAB	.03	IDTM	OEBO	03	IDTT	0EB3	03	
PTTM	OEBF	.03		%00A8	00		.0082		IEINIT=			iEXTIO.	.002A	03	
	70089	01		%008B	01	IFCP	0079	03	IFETCH=			IFPA	0082	03	
FPRET.	0106		IFP1	0090		IFP1A	.00A4	03 03	IFP1B	.00AC	03 03	IFP1C	.00AF	03 03	
FP2	00B4 00F4	03	IFP2A IFP5A	00CF 00F6	03 03	IFP3 IFP6	00D5 0104	03	IFP3A IHI =	00DE 0015	03	IFP4 IKB1	00EA 11A4	03	
FP5 KB2	11AC	03 03	IFFJA	1184	<u>03</u>	IKB4	11BC	 	IKB5	1104		IKB6	1108	03	
	0020	03	ILO =		03		40002	00	INDX1A=		00		20003		
NDX2A=			····iNIT	1045			111E		INITKB	119E		INMOV	10A7	о́з	
	%00B2	01		700B3	01	IN3	1068	03	INSA	108A	03	INSB	1086	03	
N4	109C		IN5	10A3	.03	IP 7	00B8		IPINIT=	0003		IRQ	1329	93	
RQ1	1347	03	IRQ2	1352	03	IRQ3	135E	03	IRQ4	1361	. 03	IRQ4A	1379	03	!. <b></b>
RQ5	1390	03	ISCON =	005A		ISTORE=	000C		ITMOD =			ITMRO	002D	03	
	70088			7008A	0.1	<u>I.1</u>	.0008		12	.0010	03	i3	.0019	03	
5	0029	03	KBACC	0448	03	KBBREA		03	KBCHK	1EC4	03	KBCP	02EE	03	
BCPLK.	OFBF		KBCRT =		03	KBCRTL=		00	KBC_BE=			KBC_DC= KBC_EL=			•••
BC_DK≔ BC_ER≃			KBC_DR= KBC_ID=			KBC_EC≃ KBC_OF=			KBC_EK= KBC_ON=			KBC_XL=			
BCO BCO	1EC6		KBC1	1EC8		KBC2	1ED1		KBDISF	001F	i	KBDT =		оз	
BDTL =		. 00	KBENT	043E	03	KBE1	0445	03	KBFNT =		03	KBFNTL=		00	
BIDNU=			KBIN	0091		KBOUT	0090	01	KBPDT =			KBPDTL=		öö	
BSCRO	0FE3	03	KBSCR1	OFED	03	KBSCR2	OFFO	.03	KBSPAC	0436	03	KBS1	043C	03	
BTAB	041B		KB_ACC=	009F		KB_BRE=	0089		KB_CAP=			KB_CNT=	0020		
B_COM=	009D		KB_DAS=	009C		KB_DEC=	009A		KB_DOW=	.0081		KB_ENT=			
B_ERA=			KB_F1 =			KB_F2 =			KB_F3 =			KB_F4 =			
B_F5 =			KB-F6=.			KB_F7.=	0086		KB_F8_=			KB+F.9=.			
B_HEL=			KB_HOM=			KB_ID =			KB_LEF=			KB_POW=			
B_RIG=			KB-SCR=.			K₽⊸ŞET≓.			KB_SHF=			KB-SPA=			• • • • • • • • • • • • • • • • • • • •
B_TAB=			KB_UP =				0090		KB_1 =				0092		
<u>B</u> -3=			KB-4≡.			K₽"5≘.			KB-A=.			K₽.,7≡			•••••
B_8 =	0098		KB_9 =	0099		KC1	OFC8	03	KPA	OBCD	03	KPADAL	0031	01	

.KPR	02F3	03	KPR2	. 0413	оз	KP0	.02F4	03	KP1	0303	03	KP10	озсв		
KP11	0384	03	KP12	03 <b>B4</b>	03	KP13	03A3	03	KP14	03A1	03	KP15	03B2	03	
.KP.16	0365	03	KP16A	036B	03	KP2	030F	03	КРЗ	0335	03	KP4	034B	Q3	
KP5	0345	03	KP6	034F	03	KP7	0355	03	KP8	038A	03	KP9	0371	03	•••••
.KSR	OFFE.	03	KTM1	0421	03	KTM2		93	KTQ	Q42A	03	KT.1	. 042D	03.	
LCIF	021B	03		000A		LINADH=			LINADL=			LII	0221		
LKB	02E8	03	LKBR	02ED	03	LKUP1	13F8	03	LNOFF	1BC9	03	LNON	1BCC	03	
LN1B1	1407	93	LN1B2	14CF	<u>63</u>	LN3B	1648	03	LN3B1	164D		LN3B10	166B	<u></u> 03	•• • • • • • • • • • • • • • • • • • • •
LN3B11	1670	03	LN3B12	1675	03	LN3B13	167A	03	LN3B14	167F	03				
LN3B5	1657		LN3B7	1650	ŏ3	LN3B8	1661	93	LN3B9		<u>9</u> 9	LN3B3	1652	03.	
LN3C2	1689	03	LN3C3	168E	03	LN3C4	1693	03		1666		LN3C1	1684	03	
LN3G1	1660	03	LN3C3 LN3G2		03				LN3E1	1698	03	LN3E2	1490	03	
	177D			16A8		LN4B1	1769	03	LN4B2	176F	03	LN4D1	1775	03	
LN4D2		03	LN4D3	1785	03	LN4D4	.178D	93	LN7D1	19B7	03	LŅZD2	1900	03	
LN7D3	1909	03	LN7D4	19D2	03	LN7F1	19DB	03	LN7F2	19DC	03	LN8B1	1AE4	03	
LN8B2	1AED	03	LOOKUP	13E9	03	LORDER=			LOT=	1432		LOTL =	= 0013	00	
L1	146E	03	L1A	149E	03	L1T =	146B	03	L1TL =		00	L2	150D	03	
.L2A	1516	03	L2B	151A	03	L2C	152B	03	L2D	1538	03	L2T <del>.</del> L3	= 1501	οз	
L2TL :	= 0004	00	L25EN	0015	01	L25MEM=	5780		L250N	0037	01	L3	1598		
L3A	1500	03	L3AT =	161E	03	L3ATL =	: 000E	00	L3B	15E7	03	L3C	1600	03	
L3D	1610	03	L3T =	158C	оз		0004	öö	L4	16F8	ō3	L4A	1717	оŝ	• • • • • • • • • • • • • • • • • • • •
L4B	173B	03	L4T =	16EC	03	L4TL =		00	L5	1808	03	L5T =		03	
LSTL ::			L6	1897	ŏš	E6T =	: 188B			0004		L.	193F	03	• • • • • • • • • • • • • • • • • • •
L7C1	198D	03	L7C2	1996	03	L7D	19AE	03	L7T =						
Ľġ	i ASC	ŏš	LSA	1AB0	3		1A74	03			03	L7TL=	. 0004	00.	
	= 1A80	03	MAIN	0036	03			0.3		0004	00		= 0004	QQ	
MNI	0042	63				MAXCHA=			MAXLIN=			MLATCH=			*
			MN1A	004E		MN1B	0051	03	MN1C	005D	03	MN1D	0062	03	
.MN2	0065.	03	MN3	.006E	03	MODE =			MONITO	003A	01	MSGCL	1E94	03. 03	
MSGIC	1EB3	03	MSGOL	1EA5	03	NKC	OB2D	03	NORM	0087	03	NRMO	0093	03	
NRM1	009A	03	NULL =	0000		NUMCHA=			NUML I N=			NVRCHK	1F22	03	
NVRSUM:			NVR1	1F27	03	OBUF =	57F0		OFETCH=	0010		OKTRAN	0013	01	
OLEN =	= 57F7		ONLINE	002E	01	OSTORE=	000F		OV	%00D2	01	Р	%00D0	01	
PBS =	- ocap.	03	PBS1	0042	03	PCIF	OICA	оз	PCKB	026D	03	PĊLK	OB7A		•• •• • • • • • • • • • • • • • • • • •
PCL1A	OB7F	03	PCL1B	0B88	03	PCL1C	OB91	03	PCL2	0B95	03	PCOF	022A	03	
PCR	~~055C~	93	·····PCRLF···	1054F	93	PDC	ODF4	ö <u>.</u>	PDC1	0E11	<u>.</u> 63	PĎČŽ	OE31		• • • • • • • • • • • • • • • • • • • •
PDL	0D88	03	PDLA	OD95	03	PDLO	OD9B	03	PDL1	OD9F	03	PDL2	ODA9	03	
PFIELD:			ฮ์เรีย	∵ŏŏčĭ∵	3	Piči	ODDS	გვ	รับโร้	·· 0D47	63	·····PILA····	OD52		
PILRET	0D72	03	PILRT1	OD75	03	PILO									
PIR	···oifi	გვ	·····PII	01DD	3	PI2	OD5C	03	PIL1	OD60	03	PIL2	OD6A	03	
PKL	0291	03					01E4			O1FO	63	PKDLY	0285	03	
PK2		03	PKR	02B3	03	PK0	027B	03	PK1	.027F	03	<u>PK1</u> A	028F	03	
	0294		PK3	02A5	03	PK4	02A9	03	PK5	02B0	03	PLF	0561	03	
PLFCR	0555	03	PLFLP	0588	03	PLFRET	0594	03	PLF1	0567	03	PLF2	056D	.03	
PLF3	0571	03	PMADRH≔			PMADRL=			PMBUF =	57F8		FMNUM =	001B		
PMVALU=			PORTF	0002	01	P01	022F	03	P02	0234	03	P03	023F	03	
PRINT	1002	03	PRLF	0518	Q3	PRNTE	~6623~	oi	PRNT1	1DE4		····PRNT2	1E04	03	• • • • • • • • • • • • • • • • • • • •
PRNT3	1E0E	03	PROTEC	000E	01	PRTF2	0030	01	PRTYEN	0028	01	PRTYEV	002A	01	
PRTYST	~0029~	01	P\$	%00BC	01	PSD	0971	öä	PSDB	0982		PSDB1	0985	·őŝ·	•• • • • • • • • • • • • • • • • • • • •
PSDB2	098A	03	PSDE	09A2	03	PSDF	0018	01	PSDRET	09CF	03	PSD1	0200 02A0	03	
PSD2	09B7	งัฐ	หรับรั	09BF	<u>ŏ</u> š	PSD4	09CD		PSOF	OE6D	63			00	
PTR1	20000		PTR1A =		00		%0001	03			03	PSW DTO	%00D0		
PT1	%000BB	oi	···· PUTCHA		os			88	PTR2A =			PTO	700B9		
	%00BB	01				P01	0170	03	PU2	0186	03	PXO	%00B8	oi.	
PX1				70080	00		70090	00	P2	%00A0	00	P3	%00B0	00	
RAMCHK.	TEFT.	03	RAMCON	1FD1	93	RAMERR	1F1E	03	RAMP	OAFA	03	RAMO	1EF6	юз	
RAM1	1EFA	03	RAM2	1F1D	03		%009A	01	RD	%00B7	01	REN	%009C	01	
REPF	0034	oi	RGL	066F		RGR	067A	03	ŔĬ	70098	01	RING	066B	03	
RLL	0537	03	RLR	0543	03	RL1	0548	03	RL1A	054C	03	RMC1	1EE2	03	
RMC2	TÉFO.	3	RMS	.092C	03	RMSRET	0943			0956	ŏš	RMSTL =			••••••
RMS1	0938	03	ROMCHK	1ED9	03		%00D3	01		%00D4	01			01	
					<del>.</del>					ALCOUNT		RTS	0095		

Z29ROM; MICROBENCH 80 SYMBOL TABLE	51 CROSS ASSEMBLER (VI	)-218 24-FEB-83 15:30:06	PAGE 244+		
RT1 OBOA O3	RUBOUT= 007F	RVVA 0008 01	RXD %00B0 01	R2_BAC= 1024	
R2_DCL= 103F	R2_DIS= 1042	R2_ESC= 102D	R2_FIL= 100C	R2_ICU= 101B	
R2_IKB= 101E	R2_IN = 1000	R2_IN2= 1003	R2_IRQ= 100F	R2_KY1= 1027	
R2_KY2= 102A	R2_PRN= 103C	R2_SER= 1012	R2_SET= 1021 R2_TP = 1036	R2_STA= 1006 R2_T25= 1039	
R2_ST0= 1009 R2_XMT= 1015	R2_TC = 1030 R2_XON= 1018	R2_TL = 1033 SAV2 1B0D 03	SBC 0B33 03	SBLR 0F96 03	• • • • • • • • • • • • • • • • • • • •
SBR 0E4B 03	SBRET DESE 03	SBRT OFAF 03	SBRO OE5A O3	SBUF %0099 00	
SB1 OF9C 03	SB2 0FA6 03	SB3 0FA7 03	SCH 0036 03	SCLRL 1B2E 03	
SCLRLR 1B2C 03	SCON %0098 00	SCP OEE8 03	SCRNSA 003B 01	SCRN1 = 004F	
SCRN2A= 0098	SCRN2B= OODC	SCRN3 = 0099	SCRN4A= 004E	SCRN4B= 004F	
SC1 1166 03	SC2 1171 03 SETCF 117B 03	SD1 11F0 03 SETCRT 1152 03	SERIAL 12CB 03 SETDIS 082A 03	SERIN 12DD 03 SETLCK 0093 01	• • • • • • • • • • • • • • • • • • • •
SEROUT 1313 03 SETNOR 082F 03	SETCF 117B 03 SETTAB 05A2 03	SETUP 1417 03	SETUPR 1AF6 03	SETUPS 1802 03	
SETUPO 141B 03	SETUP1 141E 03	SETUP2 14D7 03	SETUP3 1542 03	SETUP4 16B0 03	
SETUP5 1795 03	SETUP6 1865 03	SETUP7 18F4 03	SETUP8 19DD 03	SFA 126F 03	
SFB 12A5 03	SFCKB 1407 03	SFCKB1 140F 03	SFCKB2 1416 03	SFILL 1B21 03	
SFIL1 1B23 03	SFIL2 1B25 03	SFL 12AC 03	SF1 1286 03	SF2 128D 03	
SF3 12AB 03	SF4 12B9 03	SF5 12BF 03	SF6 12C7 03	SHIFTF 001C 01	
SIO 12ED 03 SLP 1840 03	SI1 12F3 03 SMS 08E7 03	SI1A 12FF 03 SMSRET 08FE 03	SI2 1301 03 SMST = 0911 03	SI3 1306 03 SMSTL = 0009 00	
	SMS 08E7 03 SMO %009F 01	SMSRET 08FE 03 SM1 %009E 01	SM2 %009D 01	SOFTIN 10E4 03	
SMS1 08F3 03 SOFTI1 111B 03	SOR 0E7F 03	SP %0081 00	SPF 01A9 03	SPO 01AE 03	
SP1 01C0 03	SP2 01C4 03	SP3 01C6 03	SRET 1AFF 03	SRT 1317 03	
SRT1 1326 03	STAB 013A 03	STACK = 005F	START 0030 03	STARTD 11DC 03	
STB1 05B0 03	STB2 05B9 03	STCALL 083A 03	STFILL 1262 03	STJMP 0836 03	
STOPDM 11FD 03	STP1A 1420 03	STP1B 142C 03	STP2A 14D9 03	STP2B 14DB 03	
STP2C14E793	STP2C1 14EC 03	STP2D 14EE 03	STP2D1 14F3 03	STP2E 14F5 03 STP3B1 1562 03	
STP2F 14F9 03	STP2F1 14FB 03 STP3C1 1572 03	STP3A 1544 03 STP3C2 157C 03	STP3B 1550 03 STP3C3 1580 03	STP3B1 1362 03 STP3D 1584 03	
STP3C 1567 03 STP3E 1588 03	STP3C1 1572 03 STP4A 16B2 03	STP3C2 157C 03 STP4B 16BE 03	STP4C 16C3 03	STP4C1 16CF 03	
STP4C2 16D9 03	STP4D 16DB 03	STP4D1 16E3 03	STP4E 16E8 03	STP5A 1798 03	
STP5B 17A4 03	STP5B1 17C1 03	STP5B2 17C4 03	STP5B3 17E5 03	STP5C 17F0 03	
STP5D17F403	STP5E17F803	STP6A184703	STP6B187503	STP6C 1879 03	
STP6D 187D 03	STP6E 1884 03	STP7A 18F6 03	STP7B 1904 03	STP7C 1908 03	
STP7D190C93	\$TPZD1191603	STP7D2191C03	STP7D3 1923 03 STP8B1 1A08 03	STP7D4 1927 03 STP8B3 1A1D 03	
STP7E 192C 03 STP8C 1A21 03	STP8A 19DF 03 STP8D 1A32 03	STP8B 19ED 03 STP8E 1A3A 03	STP8B1 1A08 03 STP8E0 1A41 03	STP8E1 1A61 03	
STP8C 1A21 03 STRET 083C 03	ST1 013B 03	ST2 014F 03	SUC 0B37 03	SUC1 0B39 03	
SXC11D9D03		SXMTL 1DAS 03	SXON1 13E2 03	SXR 13E1 03	
TAB 05C1 03	TABTAB= 0029	TBL1 05D6 03	TBL2 05DE 03	TBR 05FC 03	
TB1	TB205ED03	TB3Q5EAQ3	TB405E703	TB5Q5F7Q3	
TB6 05DC 03	TB8 %009B 01	TCNT = 003A	TCON %0088 00	TEMP %0005	
IEMPA.=.0005	TERMIN11CB93 TH1 %008D 00	TFO %008D 01 TI %0099 01	TF1 %008F 01 TLO %008A 00	THOUR = QO3D TL1 %008B 00	• • • • • • • • • • • • • • • • • • • •
THO %008C 00	TH1 %008D 00 TMOD%0089QQ	TRA 13AC 03	TRANS 139E 03	TRANSM 139B 03	
	TROK1 13BA 03	TROK2 13BD 03	TRW 13AE 03	TRO %008C 01	
TR1 %008E 01	TSCROL= 0017	TSEC = QO3B	TXD %00B1 01	TO %00B4 01	
T1 %00B5 01	UNDEF 0767 03	UNDLNA 0009 01	USCP OEF4 03	USCP1 OEFF 03	
USCP2OFO103	UXOFFS 0011 01	V\$P007201	YSPF002401	WEOL OAFD 03	
WLATCH 1E18 03	WORK %0004	WORKA = 0004	WORK1 %0006	WORK1A= 0006	
WORK2 %0007	WORK2A= 0007	WR 700B6 01	XACR 0B65 03	XALF 0868 03 XGM 0F3C 03	
XATR OF26 03	XBLINA 09F4 03	XERM 0B77 03 XICM 0B58 03	XGATM OB71 03 XKAM OB51 03	XGM OF3C O3 XKSM OB4B O3	
XHSM 0B43 03 XLATCH= 0020	XHSM1 0B45 03 XLONOF 1BBF 03	XLON1 1BC7 03	XLO 1BF5 03	XLOA 1C14 03	•••••
XL11C1D03	XL1A 1C24 03	XL10 1CBA 03	XL10A 1CBF 03	XL11 1001 03	
XL12 1CCB 03	XL12A 1CDO 03	XL13 1CD2 03	XL14 1CDC 03	XL14A 1CE1 03	
ALIE IOOD OO					

Z29ROM: SYMBOL						-218 24-FE				•••••					• • • • • • • • • • • • • • • • • • • •	
											• • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
XL18 XL2	1035 1026	03 03	XL18B XL20	. 1D4B 1D63	03	XL180			XL18D	1059	<u>03</u>	XL18E	.1D60			
	1C57	03	XL4	.1063 105D	Q3	XL3 XL4A	1028 .1074	03 03	XL3A XL4AA	1C44	03 03	XL3B	1C4B	03		
XL4C	1089	03	XL4SET			XL7	108D	03		10A0	93 03	XL4B XL8A	.,107.D 10AE	93		• • • • • • • • • • • • • • • • • • • •
	1CBQ	Q3	XML2	. 1D7A	03	XMP.1	.1DB2	03		IDBB.	03	XMP3	1DCO	03		
XMTC	1BE4	03	XMTL	1D74	03	XMTL1	1D75	03	XMTP	1DAC	03	XMTS	1BCF	03		
XMTS1		03	XMT\$2 XOFFRC	1845 0016	Q3	XMTXOF		03	XMTXQN		93	XMI25				
XPQ\$≢			XRVM	OF36.	03	XOFFSE XR2_DI	0010 0B23	01 03		0011 1B9D	03	XPCKB XSTAB	0B2A	03		
XSTABX		03	XSTAT	1BA8	ŏã	XSTATX		03		1BAE	<u>ү.ү</u>	XSTR	.09F.7 1BB8	03		• • • • • • • • • • • • • • • • • • • •
		Q3	XXPQS.=			XYPQS.=			YPOS=	0012		ZAR	.0F25.	00		
ZATR	0F08	03	ZATRO	0F14	03	ZDS	0041	01	ZDSESC	06B8	03	ZDSSBR	0E3D	03		
ZDST ZE2	VONE	Q3 03	ZDSO ZTAB	OE49 OED3	03 03	ZE1A	.06C1	03		0606	<u>03</u>	ZE1Ç	.0600	03	. <i>.</i>	
		03	ZIHD	OEDS		ZTABT =		03	ZTABTL=		00					
\$\$\$DSG	0000	00	•••••			• • • • • • • • • • • • • • • • • • • •						• • • • • • • • • • • • • • • • • • • •	• • • • • • • • •			• • • • • • • • • • • • • • • • • • • •
<b>\$\$\$</b> BSG.		01														
\$\$\$XSG		02														
	2999	છર્ <u>3</u>	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • •												
MODULE:		<b>"</b>														
HWYYYA	. 부두 / 두 두 년	+H1V	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • •	• • • • • • • • • •						• • • • • • • • •			• • • • • • • • • • • • • • • • • • • •		
FREE B: Z29, E	ORE: 10 3:Z29=A	574, WOR :Z29/E:A	DS BS/E:LC/C	•••••												
************			• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • •		• • • • • • • • •	• • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • •		
														• • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •
• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •													
			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • •					• • • • • • • • • • • • • • • • • • • •	• • • • • • • • •	• • • • • • • • • •		• • • • • • • •	• • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
																• • • • • • • • • • • • • • • • • • • •
• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •					• • • • • • • • • • • • • • • • • • • •							• • • • • • • • • • • • •	
												• • • • • • • • • • • • • • • • • • • •				
,			• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	· · · · · · · · · · · · · · ·									
			• • • • • • • • • • • • • • • • • • • •										· · · · · · · · · · ·	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
· · · · · · · · · · · · · · · · · · ·		· · · · · · · · · · · · · · · ·														
														• • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
	• • • • • • • • • •															
			• • • • • • • • • • • • • • • • • • • •	•••••	• • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •							
			• • • • • • • • • • • • • • • • • • • •						• • • • • • • • • • • • • • • • • • • •	• • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
				<b></b> .												
							• •									

ZZ9ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE S-1 CROSS REFERENCE TABLE (CREF V01-05 )

\$ 	19-16 60-38	19-36 63-11	19-53 64-4	20-7 65-10	20-8 65-48	20-19 67-11	39-9 68-46	39-38 71-32	40-13 71-49	40-36 72-30	41-16 72-47	42-45 73-32	43-11 73-40	43-40 74-32
	74-40	75-32	75-43	76-30	76-41	78-31	79-3	81-30	82-2	91-11	91-34	117-36	117-37	134-22
	135-6 194-9	140-29 194-23	140-40 197-30	149-43 197-44	. 182-39 . 199-54	183-41 200-11	. 183-50 . 203-12	. 183-55 . 203-26	186-2 207-49	186-16 208-9	189-1 <b>4</b> 208-19_	189-28 1208-34	190-46 215-28	191-43
A1M	67-17	73-14#	177 00	177 44	1// 04	200 11								
AIRM	43-38	76-13#	76-20								• • • • • • • • • • • • • •			
AIRT AIRTL	76-16 76-15	76-30# 76-41#	76-41				• • • • • • • • • • • • •							
AISM	73-35	75-13#	75-20											
ATSR	74-18	75-14	75-18	75-22#	76-14	76-18			•••••					• • • • • • • • • • • • • • • • • • • •
. A1ST . A1STL	75-16 75-15	75-32# ************************************	75-43											
A2M	.67-14	74-14#												
AZRM	74-38	81-13#.	81-20						• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •
A2SM ACA1	. 74-35 ∵87-19∵	78−13# 87-22#	78-20											
ACA2	87-17	87-23#												
ACA3	87-20	87-22	87-25#	• • • • • • • • • • • • • • • • • • • •							•••••			
ACA3A ACA5	87-28 87-26	87-29# 87-29	87-31#											
ACCESS	9-44#	33-22		38-35*										
	122-50#	122-51												
ACD1 ACDN	122 <b>-4</b> 6 67-23	122-49# 122-45#												
ACHR2	10-32#	160-8	207-9*	207-10	209-3									
	118-19#	118-20												
ACLO	118-15 67-29	118-18# 118-14#												
ACP1	94-23	94-28	94-33#											
ACP1A	94-36	94-37#												
ACP2	94-37 94-18	94-39# 94-20	94-25	94-31#	94-38	94-40								
ACPR	68-14	94-15#												
	119-42#	119-43												
ACR1 ACRT	119-38 67-26	119-41# 119-37#							• • • • • • • • • • • • • • • • • • • •					
ACU	121-47#	121-48												
	121-43	121-46#												
ACUP	67-20 67-53	121-42#. 139-15#	• • • • • • • • • • • • • • • • • • • •						• • • • • • • • • • • • • • • • • • • •				• • • • • • • • • • • • • • • • • • • •	
ADAM	139-17	139-20#					,							
ADM3	10-42# 57-22	18-4	33-51	34-37	34-52	35-48	57-35	174-34	195-7					
ADM3ES AESC1	3/542 58-19#	57-33 59-16	58-38#		• • • • • • • • • • • • • • • • • • • •				• • • • • • • • • • • • • • • • • • • •				• • • • • • • • • • • • • • • • • • • •	
ALLI1	161-26#	161-29												
ALLINI ALTCHA	156-11 9-19#	161-13# 18-35	1/12_21*	140-98	207-118	220-408	221-14	221_25	221-40	221-50	222-0			
ALTGRP	9-12#. 9-22#	<u>10739</u> 18-23	<del>14</del> 9.791% 18-28	160-9*	+847+4.%.			+6+.764		221-50	223-8		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •
ANSI	10-40#	1.7-57		34-6	34-43	35-1	35-25	36-4	37-17	57-29	195-3	221-14	221-38	222-12
ANSIES	222-20 57-14	223-13 57-17	223-25 57-19	223-41 57-22	226-14 57-25	58-14#								
ANSIT	.9/.7#9 58-17	9/54/ 65-10#	97.TA.? 65-48	94.7 <del>44</del>	×(+×	YM.A7#.			• • • • • • • • • • • • • • • • • • • •					
		45::48#.												
APBT	67-50 122-14	123-13#	123-20											

			ROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE S-2 EF.VQ1-05)
AFCA	67-32	67-56	87-16#
.APDC	67-47	133-14#	······································
APDCO	133-15	133-18#	
	.133-12#.	133-20	
APDL APDLO	67-44 129-15	129-14# 129-18#	
	129-19#	129-20	
APIC	68-44	131-14#	
APICO		131-18#	
APICI APIL	131-19# 67-41	131-20 127-14#	
APILO		127-14#	
	127-19#	127-20	
APR	94-16	95-16	95-18# 95-39 96-18
APRNT	65-46	95-13#	
ARAMP	65-37 68-8	68-35 72-13#1	96-17# 
ARMT	72-16	72-30#	THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDANCE OF THE ATTENDAN
ARMTL	72-15	72-47#	/2-4/
ASBR	68-32	95-35#	
ASBR1	95-36	95-38#	
ASBR2	95-38 68-23	95-39# 141-32#	•••••••••••••••••••••••••••••••••••••••
ASG0	90-19	90-21#	
ASG1	90-16#	98-22	
ASGM	68-11	90-14#	91-34
ASSMT	90-17	91-11#	91-34
ASGMTL ASGO	90-16 90-26	91-3 <b>4#</b> 90-25#	
ASM	68-5°	71-13#	71-20
ASMR	71-14	71-18	71-20 71-22# 72-14 72-18 73-18
ASMT	71-16	71-32#	71-49
ASMTL''' ASPF	71-15 68-41	71-49#	
ASPF1	92-20#	92-18# 92-34	•••••••••••••••••••••••••••••••••••••••
ASPF1A	92-22	92-24#	
ASPF1B		92-26#	
ASPF1C		92-28#	
ASPF1D		∵'92-30#` 92-32#	
	∵92-32…	…92-34#…	•••••••••••••••••••••••••••••••••••••••
ASPF2	92-21	92-36#	
AT2	54-52	54-54#	
ATEC 'ATRANL'	. 54-50# ```&∸31#``	68-2 22-22	"88188""1388"88""17481891
ATRIBM	6-9#	207-2	23-24 130-29 132-27 132-50 170-32 172-28 173-6 213-25 220-34 238-23
ATTRIB		22-23	143-49* 149-24* 149-30* 168-15* 206-48 206-51* 207-1* 220-18 220-25* 220-36* 224-35 224-38*
AUSCP	68-26	142-35#	
	11-12#		174-22* 175-52* 175-53
AUTOCR 'AUTOLF'	10-/# 10-8#	47-34 47-15	107-14* 107-32* 202-24* 203-48 108-14* 108-32* 202-30* 204-4
AUTOWR		22-32	97-15* 97-35* 197-14* 198-15
BANKO	```2∸27¥``	159-14	
BANK1		170-39	
BAUDRA		134-45*	
BELL BFIX	2-4# 8-23#1	20-4 17-33	56-21 46-18 46-22 46-28 49-20 49-32 122-23 125-36* 126-20 126-24* 126-25 126-36* 126-41*
+ A	0 20#	17-00	46-18 46-22 46-28 49-20 49-32 122-23 125-36* 126-20 126-24* 126-25 126-36* 126-41*

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE S-3 CROSS REFERENCE TABLE (CREF V01-05 )

128-17														
BLINKA 9-17# 143-25* 144-49* 222-41 BLINKF 9-6# 150-28* 175-31* 175-38* BLNKCN 11-10# 150-33* 175-31* 175-36 175-37* BLNKRA 11-9# 150-20* 175-32 BRI 153-18# 153-22 153-23 BR2 153-19 153-20 153-22# BRYAB 134-42 135-6# BS 2-5# 19-41 20-10 214-34 BYAB 53-15# 60-19 123-18 BYAB 53-15# 60-19 123-18 BYAB 53-20# 53-27 BYL 53-26 53-27# BYL 53-26 53-27# BYL 53-16 53-30# CAN 2-11# 17-21 CAPLOC 10-20# 152-14* 152-16 162-21 162-31* 167-18 234-27 CBA 93-15# 94-93 252-26 232-40 232-52 CCRS 5-13# 166-16													•••	
BLINKF 9-6# 150-28* 175-38*  BLNKCN 11-10# 150-33* 175-31* 175-36 175-37*  BLNKRA 11-9# 150-20* 175-32  BRI 153-16# 153-22 153-23  BR2 153-19 153-20 153-22#  BR7AB 134-42 135-6#  BS 2-5# 19-41 20-10 214-34  BTAB 53-15# 60-19 123-18  BTL 53-20# 53-27  BTL 53-26 53-27#  BTL 53-26 53-27#  BTL 53-26 53-27#  CAN 2-11# 17-21  CAPLOC 10-20# 152-14* 152-16 162-21 162-31* 167-18 234-27  CBA 93-15# 94-33 232-26 232-40 232-52  CCRSR 5-13# 166-16  CDN 61-16 63-20 120-18 122-15# 122-50					48-14	49-48	52-56	53-30	55-43	86-8	225-27	228-24	230-39	
BLNKRA 11-9# 150-20* 175-32 BR1 153-18# 153-22 153-23 BR2 153-19 153-20 153-22# BR7AB 134-42 135-6# BS 2-5# 19-41 20-10 214-34 BTAB 53-15# 60-19 123-18 BTL 53-20# 53-27# BTL 53-20# 53-27# BTR 53-16 53-30# CAN 2-11# 17-21 CAPLOC 10-20# 152-14* 152-16 162-21 162-31* 167-18 234-27 CBA 93-15# 94-33 232-26 232-40 232-52 CCRSR 5-13# 166-16 CDN 61-16 63-20 120-18 122-15# 122-50				222 71										
BR1 153-18# 153-22 153-23 BR2 153-19 153-20 153-22# BRTAB 134-42 135-6# BS 2-5# 19-41 20-10 214-34 BTAB 53-15# 60-19 123-18 BTL 53-20# 53-27 BTL 53-26 53-27# BTR 53-16 53-30# CAN 2-11# 17-21 CAPLOC 10-20# 152-14* 152-16 162-21 162-31* 167-18 234-27 CBA 93-15# 94-33 232-26 232-40 232-52 CCRSR 5-13# 166-16 CDN 61-16 63-20 120-18 122-15# 122-50				175-36	175-37*									
BR2 153-19 153-20 153-22#  BRTAB 134-42 135-6#  BS 2-5# 19-41 20-10 214-34  BTAB 53-15# 60-19 123-18  BTL 53-20# 53-27  BTL 53-26 53-27#  BTR 53-16 53-30#  CAN 2-11# 17-21  CAPLOC 10-20# 152-14* 152-16 162-21 162-31* 167-18 234-27  CBA 93-15# 94-33 232-26 232-40 232-52  CCRSR 5-13# 166-16  CDN 61-16 63-20 120-18 122-15# 122-50													•••	
BS 2-5# 19-41 20-10 214-34  BTAB 53-15# 60-19 123-18  BTL 53-20# 53-27  BTL1 53-26 53-27#  BTR 53-16 53-30#  CAN 2-11# 17-21  CAPLOC 10-20# 152-14* 152-16 162-21 162-31* 167-18 234-27  CBA 93-15# 94-33 232-26 232-40 232-52  CCRSR 5-13# 166-16  CDN 61-16 63-20 120-18 122-15# 122-50														
BTAB 53-15# 60-19 123-18 BTL 53-20# 53-27 BTL 53-26 53-27# BTL 53-26 53-27# BTR 53-16 53-30# CAN 2-11# 17-21 CAPLOC 10-20# 152-14* 152-16 162-21 162-31* 167-18 234-27 CBA 93-15# 94-33 232-26 232-40 232-52 CCRSR 5-13# 166-16 CDN 61-16 63-20 120-18 122-15# 122-50														
BTL 53-20# 53-27 BTL1 53-26 53-27# BTR 53-16 53-30# CAN 2-11# 17-21 CAPLOC 10-20# 152-14* 152-16 162-21 162-31* 167-18 234-27 CBA 93-15# 94-33 232-26 232-40 232-52 CCRSR 5-13# 166-16 CDN 61-16 63-20 120-18 122-15# 122-50				214-34										
BTL1 53-26 53-27# BTR 53-16 53-30#  CAN 2-11# 17-21  CAPLOC 10-20# 152-14* 152-16 162-21 162-31* 167-18 234-27  CBA 93-15# 94-33 232-26 232-40 232-52  CCRSR 5-13# 166-16  CDN 61-16 63-20 120-18 122-15# 122-50	BTL 53-20#	53-27												
CAN 2-11# 17-21 CAPLOC 10-20# 152-14* 152-16 162-21 162-31* 167-18 234-27 CBA 93-15# 94-33 232-26 232-40 232-52 CCRSR 5-13# 166-16 CDN 61-16 63-20 120-18 122-15# 122-50														
CAPLOC 10-20# 152-14* 152-16 162-21 162-31* 167-18 234-27 CBA 93-15# 94-33 232-26 232-40 232-52 CCRSR 5-13# 166-16 CDN 61-16 63-20 120-18 122-15# 122-50		33-30# 17-21										• • • • • • • • • • • • • • • • • • • •		
CCRSR 5-13# 166-16 CDN 61-16 63-20 120-18 122-15# 122-50	CAPLOC 10-20#	152-14*			162-31*	167-18	234-27						<b></b>	
CBN 61-16 63-20 120-18 122-15# 122-50			232-26	232-40	232-52									
CDN1 122-16 122-19#			120-18	122-15#	122-50				• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
CDN2 122-19 122-22# CDN3 122-23 122-26#														
CHARME 6-11# 44-24 45-25 206-31 207-32 238-15 242-16			45-25	206-31	207-32	238-15	242-15	242-16				• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
CHKS 243-9# CHRORL 6-33# 22-25 130-35 132-44 170-29 170-34 173-18 220-37 238-31				· · · · · · · · · · · · · · · · · · ·	· · • • • • • • • • • • • • • • • • • •									
CHRORL 6-33# 22-25 130-35 132-44 170-29 170-34 173-18 220-37 238-31 CINT 5-14# 163-39			130-33	132-44	170-29	170-34	1/3-18	220-37	238-31					
CLA 23-22 45-15# 112-36 114-16 116-50 130-22 132-22 220-30		45-15#	112-36	114-16	116-50	130-22	132-22	220-30				• • • • • • • • • • • • • • • • • • • •	••••••	
CLFO 63-26 113-44# CLFT 61-18 117-37# 118-19 118-40			119-19	119-40									•••	
CLKF 10-18# 100-14* 100-34* 167-42 199-43* 201-8					199-43*	201-8								
CLNNON 11-44# 103-21* 103-37*		163-21*	163-32*	163-37*										
CLRFLG 9-50# CLRL 112-16 112-35# 113-17 113-23 114-39 115-50 116-20		112-35#	113-17	113-23	114-39	115-50	116-20						• • • • • • • • • • • • • • • • • • • •	
CLRLIN 46-20 49-25 99-18 99-39 112-14# 126-27 128-24 161-41 162-44 213-21	CLRLIN 46-20	49-25		99-39	112-14#	126-27	128-24		162-44	. 213-21				
				75-38	76-36	88-23								
CLRTB 50-13# 54-33 140-33 CLRTB 50-17# 50-19									• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	
CMPORT 193-24 230-19 230-43 231-13 231-13# 240-32 241-35					231-13#.	240-32	241-35							
CONTLF 9-43# 33-20* 36-26 37-45 38-14 CPF 23-14# 24-19 220-22														
CPF1 23-15 23-19#	CPF1 23-15	23-19#												
CPF2 23-28 23-31# CPFR1 23-16# 23-29 23-32														
CPFR2 23-30 23-34#			and dan											
CPR 62-3 64-2 137-14#	CPR 62-3		137-14#											
CPRO 137-16 137-17# CPROA 137-17 137-19#														
CPR1 137-14 137-24#	CPR1 137-14													
CPR2 136-19 136-26 137-19 137-22 137-28 137-31# CPRM 137-24 137-33#			137-19	137-22	137-28	137-31#								
CR 2-8# 17-29 17-37 20-1 41-52 42-37 182-41 208-25 214-28 228-27 230-31 241-33 241-33			17-37	20-1	41-52	42-37	182-41	208-25	214-28	228-27	230-31	241-33	241-33	
CR1 166-24 166-26#														·
CR2 166-25 166-27# CRESET 5-3# 164-22										•				
CRNBK 10-27# 164-40 202-36 202-42* 202-50* 204-10	CRNBK 10-27#	164-40				204-10					· · · · · · · · · · · · · · · · · · ·		••••••	
CRT 19-22 61-17 119-14# 119-42 120-16 CRT1 119-15 119-18#			119-14#	119-42	129:16							.,		
CR(1 119-13 119-10#														

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE S-4 CROSS REFERENCE TABLE (CREF.V01-05 )
CRUL 10-17# 101-13* 101-31* 164-42 202-37 202-38* 202-41* 202-45 202-46* 202-49* 204-12 204-16
CS1 113-15 113-19#
CS2 113-22# 113-25 CSET 5-15# 163-29
CSET 5-15# 163-29 CSTOP 5-12#
CSTRT 5-11# 163-34 165-17
CTAB1 17-56 19-36# 20-7 CTAB1L 17-55 20-7#
CTAB2 18-3 19-16# 20-8
CTAB2L18-220-8# CTAB3 18-6 19-53# 20-19
CTAB3L18-520-19#
CTS 11-48# 16-30 177-29 CUA 58-41 61-39 63-56 124-13#
CUA1 124-18 124-19#
CUA2 124-19 124-21# CUA2A 124-21 124-22#
CUA3 124-22 124-24#
CUBO 124-17 124-28#
CUB1 124-31 124-34# CUC 124-15 124-16 124-38#
CUCO 124-39 124-46#
CUCOA 124-49 124-50# CUC1 124-41 124-42#
CUC1A 124-42 124-44#  CUC2 124-48 124-51#  CUP 19-19 61-15 63-17 118-42 121-15# 121-47
CUP1 121-16 121-18#
CUP1A 121-17 121-20# CUP1B 121-20 121-23#
CUR 124-26# 124-30 124-32 124-34
CUR1 124-35# 124-44 CUR2 124-24 124-50 124-52#
CURSOR 166-15# 175-26
D25L 81-33 99-37# DC78-46 106-31#
DCLK 158-22 232-20#
DCLKO 232-21 232-24# 234-23 240-31 DCLK1 232-29 232-31#
DCMDRE 6-4# 163-28 163-42 164-16 165-16 166-15
DDATRE 6-3# 164-25 166-19 DEOL 62-12 76-39 97-35#
DEOL 62-12 76-39 97-35# DISPSE 11-19# 232-20 232-24*
DISPST 158-28 162-46 210-16 234-16# 240-26 DKI 19-28 62-18 63-38 71-38 151-32#
DKI 19-28 62-18 63-38 71-38 151-32# DM1 170-20 170-39#
DM1A 170-45 170-55#
DM1B 170-46 170-48# DM1C 170-48 170-49#
DM1D 170-50 170-51#
DM1E 170-55 170-56# DM2 170-47 170-57#
DM3 170-49 170-56 171-2#
DM4 170-51 170-53 171-14# DMAANL 6-32# 170-28
DMADRH 8-3# 163-19* 169-30 171-12*
DMADRL 8-4# 163-18* 169-29 171-9*

729Rom; MYCROBENCH 8051 CROSS ASSEMBLER (VI)-218 24-FEB-83 15:30:06 PAGE S-5	
CROSS REFERENCE TABLE (CREF V01-05 )	
DMAMEM 6-24# 163-18 163-19 171-11	
DMAORL 6-34# 172-29 DMATYP 11-47# 169-22* 170-20 170-21* 172-50*	
DMLN 7-29# 170-43 170-46 170-48 170-50 170-55 170-57 171-3	
DMLNA 7-30# 163-17* 169-24 175-24* DMPTR 7-27# 171-4 171-5	
DMPTRA 7-28# DOXOFF 174-42 178-14# 230-17	
DOXON 157-1 178-41# 230-46 DSADRH 8-19# 17-26 57-17* 69-17* 69-38* 124-16*	
DSADRL 8-20# 17-25 57-16* 57-30* 57-33* 57-36* 69-18* 69-37* 124-15* DSPO 234-17 234-20#	
DSP1 234-27 234-30#	
DSP2 234-32 234-35# 	
DSPR 234-16 234-18# 	
DSRO 125-16 125-19# DSROA 125-19 125-20#	
DSR1 125-20 125-23# DSR1A 125-24 125-27#	
DSR2 125-27 125-30#	
DSRERR 125-31 125-32# DSRERR 125-30 125-34#	
DSROK 125-32 125-35# DSTARE 6-5# 163-42 164-13 164-16 175-28	
DTR 11-41# 178-19* 178-46* DX 9-31# 22-14* 166-25 166-29*	
E25L 78-34 99-15# EACR 71-47 79-1 107-14#	
EALF 78-55 108-14#	
EAROM 6-26# 159-35 211-17	
EAROMR 9-8# 159-29* 159-31* EAROMS 9-5# 211-30* 211-32*	
EB1 115-41 115-44# EB2 115-45# 115-52	······································
EB3 115-46 115-49# EBD 61-48 88-19 115-40#	
EBL 62-4 89-20 114-14# 115-42 115-47 EBLNK 91-23 144-49#	
EC 81-45 106-13# EE1 116-16# 116-21	
EE1A 116-17 116-18#	
EEL 62-1 89-24 114-38#	
EER 116-18 116-22# 116-47 EERM 71-44 110-14#	
EGATM 71-35 109-14# EGM 61-20 91-29 146-13#	
EGTT 74-21 74-32# 74-40 EGTTL 74-20 74-40#	
EHALF 63-50 91-17 144-13# EHSM 61-41 78-40 102-13# 193-50	
ÉÏCM 61-14 71-41 105-13#	
EID 67-35 88-13# EID1 88-14 88-17#	

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE S-6 CROSS REFERENCE TABLE (CREE V01-05 )	
CROSS REFERENCE TABLE (CREF. V01-05.)	•••••••••••••••••••••••••••••••••••••••
EID2 88-18 88-21#	
EIL 67-38 89-14#	
EIL1 89-15 89-18# EIL2 89-19 89-22#	
EIR 88-22 88-25# 89-23	
EKAM. 61-11. 65-28 78-52 104-13# EKC 81-36 100-14#	•••••
EKI19-2562-1663-4172-36151-13# EKSM 62-9 78-49 103-13#	
ELB65-1366-16#	
ELBI 00-22 6/-11# 68-46	
ENBLCU 9-40# 106-13* 106-31* 166-24	
EDL 61-25 63-29 89-16 115-14# 116-14 EDL1 115-16# 116-42	
EQMT 73-21 73-32# 73-40	
EQMTL 73-20 73-40# ERM 9-41# 110-14* 110-34* 168-16* 168-38* 172-23	
ERM1 240-37# 240-40	
ERM2 240-46# 240-48 ERM3 241-3# 241-5	
ERM4241-11#241-13	
ERM5 241-19# 241-21 ERM6 241-27# 241-29	
ERRO 240-25 240-29#	
ERR1 240-43 240-51# ERR2 240-51 241-8#	
ERR3 241-8 241-16# ERR4 241-16 241-24#	
ERR4 241-16 241-24# ERR5 241-24 241-32#	
ERRUR 161-43 207-40 240-24#	
ERVM 62-5 91-26 145-13#	
	42-19 42-22
221-52 222-2 222-14 222-22 223-15 223-19 223-28 223-32 223-46 224-10	221-42 221-46
ESCLOD 18-20 19-39 57-14# ESCRET 57-20 57-20 57-20 57-20	
EUNDL 91-20 144-31#	•••••
EZM 75-35 76-33 98-31# FCIF 16-22 26-15# 56-22	
FCKB 16-41 30-17 31-15# 153-18 181-15 181-18 207-27 236-18 FCOF 16-32 29-15#	
FI1 26-16 26-20#	
FI2 26-20 26-23# FI3 26-31 26-34#	
FILL 170-25 170-27#	•••••
FK1 31-17 31-21# FK2 31-33 31-36#	
FKDLY 31-28 31-29 31-37 31-40 31-46#	
FKDLY1 31-47 31-47# FKL 31-27# 31-38	
FKR 31-34 31-40#	
FL1 170-30# 170-35 F01 29-16 29-20#	
FOR 29-28 29-32#	
FREQ 10-28# 164-30 164-38 170-45 175-43 203-1* 204-24	
	***************************************

729ROM; MICROBENCH 8051 CROSS ASSEMBLER (VI)-218 24-FEB-83 15:30:06 PAGE S-7 CROSS REFERENCE TABLE (CREF VO1-05 ) 188-55* 190-25 FULLDP 10-10# 28-18 77 149-16# GODES 65-16 147-26# GOPICK 119-51 148-4 149⁻⁻20# G1DES 65-19 147-35# GIPICK 19-48 148-10 109-14* 109-34* 168-17* 220-21 GATM 9-55# 1 147-44# 147-39 GDES... 147-36 GDESO 147-45# 147-49 ...... GDES1 147-44 147-48¥ GDES2 147-48 147-51# ..... 147-55# GDES2A 147-54 GDES3 147-46 148-1# 148-7# "GDES3A"148-1" GDESR 147-53 147-55 148-12# .....GETNUM: 196-31 196-44 214-17# 85-15# 87-16 87-25 88-13 78-13 81-13 GNP 54-50 71-13 72-13 75-13 76-13 92-26 .96-17 111-13 111-17 111-21 118-14 119-37 121-42 122-45 123-13 125-15 125-23 195-35 94-15 133-14 139-15 141-32 142-35 150-16 : 127-14 129-14 226-15 131-14 GNP1 85-15 85-20# 85-15 85-16* GNP2 85-23 85-26# GNPF 9-39# 58-15* GNUMO 214-19# 214-49 GNUM1 214-24# 214-41 GNUM2 214-28 214-34# ..... GNUM2A 214-34 214-36# GNUM2B 214-35 214-40# GNUM2L 214-42# 214-43 GNUM4 214-36 214-53# ..... GNUM4A 214-53 214-55# GNUM4B 214-55 214-56# ..... GNUM4C 215-1 215-2# 215-28 GNUMR 214-29# 214-54 215-25# GNUMS 215-14 149-18 149-24# GPICK ..... GPICKT 149-28 149-43# GRPH 9-20# 18-21 146-13* 146-31* 220-42* 221-1* 221-9 9-29# GSEL 148-4 148-10 149-16* 149-20* 148-9* 149-17 149-21 148-7* GSET 8-32# 148-2* 148-3* 9-18# 168-37* 222-53 144-13* HALFIA 143-27* HAZ 10-43# 17-22 18-18 33-52 34-38 34-53 35-49 124-17 124-39 137-14 160-11 174-27 174-35 193-40 HAZESC 57-25 57-36 59-14# HAZINI 160-12 168-37# 193-41 HAZT 59-15 63-11# 59-14 HAZTL 64-4# 118-39# 20-11 HCLFT HCLFT1 118-44 118-47# HORT 20-14 120-15# 120-20 120-23# HCRT1 HEE 1 116-45# 116-54 116-46 HEE1A 116-47# HEED 63-35 116-41# HIR 168-18 168-39# 178-17 178-44 189-4* 190-33 16-29 177-28 HNDSHK 10-33# 17-35* 102-14* 154-22* 154-27* 8-31# HSLINE 102-13* 102-32* 154-14 193-47 195-14 HSMODE 10-21# 17-30 9-52# 17-36* 102-34* 154-21* 230-45 HSSTOP.... 16-21 2-6# 19-44 20-16 37-22

											<b></b> .		
Z29ROM; MICROB	ENCH 8051	CROSS ASS	EMBLER (V	/1)-218 2	4-FEB-83	15:30:06	PAGE S-8						
CROSS REFERENC	·+I П\$H\$\	SOBCYX4TX	'₽/	• • • • • • • • • • • • • • • • • • • •					• • • • • • • • • • • • • • • • • • • •				•••••
HTO 55-17	55-20#	• • • • • • • • • • • • • • • • • • • •						• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	
HT1 55-27	55-29#												
HT2 55-28 HT3 55-36	55-34# 55-41#												
HTAB 20-17	55-13#			• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
HTC 54-32# HTS 54-13#		140-32 140-38											
HTS1 54-14#		140-30											
II 15-15 I2 15-24	15-17# 15-26#											• • • • • • • • • • • • • • • •	
13 15-33	15-35#			• • • • • • • • • • • • • • • • • • • •			, ,	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
I5 15-42 IBUF 6-14#	15-44# 25-16	·····	·····	878%							· · · · · · · · · · · · · · · · · · ·		
IC1 163-52	23-16 163-55#	25-33	25-34	26-24	26-31	26-32	27-21	242-6	242-7				
ICMODE 9-51# ICOUNT 8-8#		35-3	35-12		105-31*	234-37					• • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •
ICOUNT 8-8# IDT 61-40	25-21 138-15#	25-30*	26-15	26-34*	27-16			• • • • • • • • • • • • • • • • • • • •					
IDTM 138-15 IDTT 61-55	138-18#												
IDTT 61-55 IDTTM 138-41	138-37# 138-44#												
IEDMA 2-33#		172-49		• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •							• • • • • • • • • • • • • • • • • • • •
IEINIT 2-32# IEXTIO 15-16	161-37 15-49#	170-18		• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •							• • • • • • • • • • • • • • • • • • • •	
IFCP 16-24 IFETCH 8-7#	17-21# 26-25			nnavnasii.								• • • • • • • • • • • • • • • • • • • •	
IFETCH 8-7# IFP1 17-29	26-25 17-30	26-29* 17-41#	26-30	26-32*	27-22								
IFP1A 17-43	17-44#					• • • • • • • • • • • • • • • • • • • •					••••••		•••••
IFP1B 17-41 IFP1C 17-49	17-49#			• • • • • • • • • • • • • • • • • • • •								• • • • • • • • • • • • • • • • • • • •	
IFP2 17-50	17-55#											• • • • • • • • • • • • • • • • • • • •	
IFP2A 17-57 IFP3 17-51	18-1 18-18#	18-4	18-7#										
IFP3A 18-18	18-19	18-21#		• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	
IFP4 18-22 1FP5 18-21	#-26# 18-23	18-26	18-28	18-35#									
IFP5A 18-30	18-36#					<b>.</b>							
IFP6 17-44 IFPA 17-21	17-47 17-22	18-38 17-25#	18-44#										
IFPRET 18-12	18-45#						• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	
IHI 6-16# IKBI 167-18	25-27 167-20#1			• • • • • • • • • • • • • • • • • • • •								•••	
IKB2 167-26	167-28#												
IKB3 167-34 IKB4 167-42	167-36# 167-44#												
1KB5 167-56	167-52#									• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
IKB6 167-20 ************************************	167-28 25-22	167-36 25-33	167-44	167-52	167-55#								
ILO 6-15#	26-20												
IN3 159-38# IN3A 160-1	159-49 160-3#												
IN3B159-57	160-1#	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			••••					• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
IN4 160-11 IN5 160-15	160-14# 160-17#											• • • • • • • • • • • • • • • • • • • •	
INDX1 7-11#	50-15	50-19	51-24	51-32	52-27	52-51	113-20	.113-22	113-24	. 115-44	115-49	. 115-51	116-15
116-16 122-51	116-17 127-18	116-19 127-20	116-44 129-18	116-45 129-20	116-46 130-18	116-49 130-43	118-18 132-19	118-20	119-41	119-43	121-46	121-48	122-49
159-36	159-49	160-30	161-24	161-29	186-36	187-9	211-18	132-32 211-28	132-56 212-16	153-16 212-21	153-22 213-26	159-18 213-29	

729ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE S-9 CROSS REFERENCE TABLE (CREF V01-05 ) 239-15 239-19 236-23 237-25 237-16 238-18 238-36 INDX1A 7-12# 115-46 7-13# 52-44 TNDX2 52-33 131-18 131-20 133-18 133-20 153-17 153-23 172-18 172-30 173-22 186-38 187-7 212-20 212-17 INDX2A 7-14# 236-17 236-22 237-17 237-24 238-19 238-35 INIT 159-14# 161-20 162-27 INITCR 161-33 163-14# 167-14# 182-29 199-37 199-44 161-44 INITKB 157-13 INMOV 159-20 159-25 160-25# 160-30 IPINIT 2-31# 161-31 TRO: 156-41 175-18# IRQ1 175-34 175-36# · 175-36 · 175-41# TRØ2 175-33 175-35 175-43 IRQ3 175-46# .....TRQ4 175-45 175-47# IRQ4A 175-51 175-54 176-1# 175-50 176-4 176-8 176-11# 175-44 175-46 ISCON 2-30# 159-53 25-34* 159-16 25-17 TSTORE 8-6# ITMOD 2-29# 159-52 15-25 15-50# TTMRO KBACC 38-35# 39-18 KBBREA 39-24 153-15# 236-23 KBCO 236-17# ..... KBC1 236-18# 236-20 KBC2 236-19 236-22# KBCHK 161-18 236-16# KBCP 16-43 30-19 KBCPLK 39-27 152-14# KBCRT 33-48 40-13# 33-47 KBCRTL 40-36# 4-31# KBC_BE 56-16 ..... 167-38 KBCLDC 4-33# 100-35 KBC_DK 151-33 4-38# 167-30 KBC_DR 167-46 4-37# KBC_EC 4-32# 100-15 167-38 ..... KBC_EK 167-30 4-39# 151-14 KBC_EL 4-34# 152-15 167-14 167-17 KBC_ER 4-36# 167-46 167-49 KBC_ID 4-30# 167-54 ..... KBC_OF 4-40# 167-22 KBC_ON 4-41# 167-22 ..... KBC_XL 4-35# 152-17 167-14 151-13* 151-32* 167-34 **KBDISF** 9-45# 16-40 KBDT 33-29 39-9# KBDTL 33-28 39-38# ...... KBE 1 38-14 38-17# **KBENT** 38-14# 39-36 ..... KEFNT 35-54 43-11# **KBFNTL** 35-53 43-40# ..... KBIDNU 8-30# 35-42* KBIN 11-38# 30-15 30-36 30-47 31-17 31-30 32-15 30-53* KBOUT 11-37# 30-24* 30-29* 30-45* 31-27* 31-31* KBPDT 34-25 42-45 41-16# KBPDTI 42-45# 34-24 KBS1 37-45 37-47# KBSCR1 154-16 154-19#

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE S-10 CROSS REFERENCE TABLE (CREF V01-05.)	
KBSCR2 154-14 154-21#	
KBSCRO 39-30 154-14# KBSPAC 37-44# 39-15	
KBTAB 37-15# 39-12	
KB_0 3-30# 41-18 42-3 KB_1 3-31# 41-21 42-6	
KB_2 3-32# 41-24 42-9	
KB_3 3-33# 41-27 42-12 KB_4 3-34# 41-30 42-15	••
KB_5 3-35# 41-33 42-18 KB_6 3-36# 41-36 42-21	• •
KB_7 3-37# 34-54 41-39 42-24	
KB_8 3-38# 41-42 42-27 KB_9 3-39# 41-45 42-30	
KB_ACC 4-7# 39-17 181-17	• •
KB_BRE 3-23# 39-23 153-20 KB_CAP 3-24# 39-26	• •
KB_CNT 3-11# KB_COM 3-43# 41-57 42-42	
KB_DAS 3-42# 41-54 42-39	
KB_DEC 3-40# 41-48 42-33 KB_DOW 3-15# 40-18 186-13	
KB_ENT 3-41# 38-15 39-35 41-51 42-36	••
KB_ERA 3-19# 40-33 KB_F1 4-9# 43-13	• •
KB_F2 4-10# 43-16 KB_F3 4-11# 43-19	
KB_F4 4-12# 43-22	
KB_F5 4-13# 43-25 KB_F6 4-14# 43-28	
KB_F7 4-15# 43-31	
KB_F9 4-17# 43-37	• •
KB_HEL 3-20# 40-30 KB_HOM 3-18# 40-27	• •
KB_ID 4-19# 35-39	
KB_LEF 3-16# 40-21 186-4 KB_POW 3-28# 39-20 236-20	
KB_RIG 3-17# 40-24 186-7 KB_SCR 3-21# 39-29	
KBLSET 3-22# 39-32 183-35 183-38 208-28 208-31	• •
KB_SHF 3-10# 34-54 42-3 42-6 42-9 42-12 42-15 42-18 42-21 42-24 42-27 42-30 42-33 42-36 42-39 42-39 42-42 183-38 208-31	
KB_SPA 3-26# 39-14 208-21	
KB_TAB 3-25# 39-11 KB_UP 3-14# 40-15 186-10	
KCi 152-16 152-18# KPO 33-13 33-17#	
KP1 33-22 33-28#	• • • •
KP10 35-25 35-29 35-31# KP11 34-43 34-45#	
KP12 34-54 35-19#	
KP13 35-1 35-7# KP14 35-3 35-5#	
KP15 35-12 35-14#	
KP16 34-27 34-30# KP16A 34-33# 34-37 34-38 34-52 34-53	••

	ENCH 8051	CROSS ASS	EMBLER (V	1)-218 2	4-FEB-83	15:30:06	PAGE S-11						
CROSS REFERENC												<i>.</i>	
		•											
KP2 33-31	33-42#	38-16											
KP3 33-57 KP4 34-4	34-3# 34-5	34-9	34-13#										
KP5 34-6	34-10#												
KP6 33-50	34-19#												
KP7 34-20 KP8 34-30	34-22# 34-52#		• • • • • • • • • • • • • • • • • • • •										
KP9 34-32	34-37#									· • • • • • • • • • • • • • • • • • • •			
KPA 33-23 KPADAL 10-16#	35-36# 34-32	104-128	104-218	199_20*	200-29								
KPADSH 10-15#	34-34 34-20	103-13*	104-31* 103-31*	199-24*	200-33								
KPB 35-39.	35-48#												
KPC 36-4 KPC1 36-26	36-12# 36-29#												
KPC0F 33-56	36-29# 34-2		34-14	34-34	34-41	34-45	34-47	34-56	35-5	35-8	35-10	35-14	35-22
35-27 KPR 33-15#	35-31 33-44	36-3 33-51	36-6 33-52	36-8 34-28	36-14 34-35	36-18	36-20	36-26#			· · · · · · · · · · · · · · · · · · ·		
KPR2 35-24	35-48 35-48		35-54 35-54	36-21#									
KSR 154-23	154-28#					,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,							
KT0 37-17 KT1 37-15	37-20 <b>#</b> 37-18	37-22#								• • • • • • • • • • • • • • • • • • • •			
KTM137-16.	37-25#												
KTM2 37-19 LQT 179-22	37-26#	102-41											
LOT 179-22 LOTL 179-21	#183-41	+8974	• • • • • • • • • • • • • • • • • • • •										
	184-15#												
L1A 184-21 L1T 182-21	184-23# 183-50#	183-55											
L1TL 182-20	183-55#												
L2													
						124-22	142-14	210-15	229-15	234-17			
L25EN 9-32#	16-46	87-20 213-24	99-15 217-17	99-16* 217-18	21.7-32	232-31	142-16 232-43			234-17			
L25EN 9-32# L25MEM6-12# L250N 10-22#	16-46 186-33 16-45	87-20 213-24 162-42	99-15 217-17 197-8*	.217-18 198-9	217-32 210-14	232-31 234-16	232-43	232-55					
L25EN 9-32# L25MEM 4-12# L250N 10-22# L2A 186-37#	16-46 186-33 16-45 187-9	87-20 213-24 162-42	99-15 217-17 197-8*	.217-18 198-9	217-32 210-14	232-31 234-16	232-43	232-55					
L25EN 9-32# L25MEM 6-12# L25ON 10-22# L2A 186-37# L2B 186-40# L2C 186-49	16-46 186-33 16-45 187-9 187-7 186-52#	87-20 . 213-24 162-42	99-15 217-17 197-8*	217-18 198-9	217-32 210-14	232-31 234-16	232-43	232-55					•••••••••••
L25EN 9-32# L25MEM 6-12# L25ON 10-22# L2A 184-37# L2B 186-40# L2C 186-49 L2D 186-55	16-46 186-33 16-45 187-7 187-7 186-52# 187-1	87-20 .213-24 .162-42 	99-15 217-17 197-8*	217-18 198-9	217-32 210-14	232-31 234-16	232-43	232-55				· · · · · · · · · · · · · · · · · · ·	
L25EN 9-32# L25MEM 6-12# L25ON 10-22# L2A 186-37# L2B 186-40# L2C 186-49 L2D 186-55 L2T 185-18 L2TL 185-17	16-46 .186-33. .16-45 .187-9. .187-7 .186-52# .187-1 .186-2#. .186-16#	87-20 .213-24 .162-42 	99-15 217-17 197-8*	.217-18  198-9	217-32 210-14	.232-31 234-16	232-43	. 232-55					
L25EN 9-32# L25MEM	16-46 .186-33. 16-45 .187-9. .187-7 .186-52# .187-1 .186-2# .186-16# .189-45#	87-20 .213-24 .162-42 	99-15 217-17 197-8*	.217-18  198-9	217-32 210-14	.232-31 234-16	232-43	. 232-55					
L25EN 9-32# L25MEM 6-12# L25ON 10-22# L2A 186-37# L2B 186-40# L2C 186-49 L2D 186-55 L2T 185-18 L2TL 185-17 L3 188-14 L3A 190-3	16-46 186-33 16-45 187-7 187-7 186-52# 187-1 186-2# 186-16# 189-45# 190-6#	87-20 .213-24 .162-42 	99-15 217-17 197-8*	.217-18 198-9	217-32 210-14	232-31 234-16	232-43	232-55					
L25EN 9-32# L25MEM 6-12# L25ON 10-22# L2A 186-37# L2B 186-40# L2C 186-49. L2D 186-55 L2T 185-18. L2TL 185-17 L3 188-14. L3A 190-3 L3AT 188-26. L3ATL 188-25	16-46 186-33 16-45 187-7 187-7 186-52# 187-1 186-16# 186-16# 190-6# 190-1	87-20 .213-24 .162-42 	99-15 217-17 197-8* 	217-18 198-9	217-32 210-14	232-31 234-16	232-43	232-55					
L25EN 9-32# L25MEM .4-12# L250N 10-22# L2A .184-37# L2B 186-40# L2C .186-49. L2D 186-55 L2T .185-18 L2TL 185-17 L3 .188-14 L3A 190-3 L3AT .188-26 L3ATL 188-25 L3B .190-13	16-46 186-33 16-45 187-7 187-7 186-52# 187-1 186-16# 186-16# 189-45# 190-6# 189-30 190-15	87-20 .213-24 .162-42 	99-15 217-17 197-8* 	217-18 198-9	217-32 210-14	232-31 234-16	232-43	232-55					
L25EN 9-32# L25MEM 4-12# L25ON 10-22# L2A 184-37# L2B 186-40# L2C 186-49. L2D 186-55 L2T 185-17 L3 188-14 L3A 190-3 L3AT 188-24 L3ATL 188-25 L3B 190-13 L3C 190-25 L3D 190-33	16-46 186-33 16-45 187-7 187-7 186-52# 187-1 186-16# 182-45# 190-6# 190-1 188-30 190-27# 190-27#	87-20 .213-24 .162-42 	99-15 217-17 197-8* 	217-18 198-9	217-32 210-14	232-31 234-16	232-43	232-55					
L25EN 9-32# L25MEM	16-46 186-33 16-45 187-7 187-7 186-52# 187-1 186-16# 189-45# 190-6# 190-1 188-30 190-27# 190-35# 190-35#	87-20 .213-24 .162-42 	99-15 217-17 197-8* 191-43 191-43# 190-19#	217-18 198-9	217-32 210-14	232-31 234-16	232-43	232-55					
L25EN 9-32# L25MEM 4-12# L25ON 10-22# L2A 184-37# L2B 186-40# L2C 186-49. L2D 186-55 L2T 185-17 L3 188-14 L3A 190-3 L3AT 188-24 L3ATL 188-25 L3B 190-13 L3C 190-25 L3D 190-33	16-46 186-33 16-45 187-7 187-7 186-52# 187-1 186-16# 189-45# 190-6# 190-1 188-30 190-27# 190-35# 190-35#	87-20 .213-24 .162-42 	99-15 217-17 197-8* 191-43 191-43# 190-19#	217-18 198-9	217-32 210-14	232-31 234-16	232-43	232-55					
L25EN 9-32# L25MEM	16-46 186-33 16-45 187-7 187-7 186-52# 187-1 186-16# 189-45# 190-6# 190-1 189-30 190-27# 190-27# 190-35# 189-14# 189-240# 194-40#	87-20 .213-24 .162-42 	99-15 217-17 197-8* 	217-18	217-32	232-31 234-16	232-43	232-55					
L25EN 9-32# L25MEM 4-12# L250N 10-22# L2A 184-37# L2B 186-40# L2C 186-49. L2D 186-55 L2T 185-17 L3 188-14 L3A 190-3 L3AT 188-25 L3AT 188-25 L3B 190-13 L3C 190-25 L3D 190-33 L3T 188-17 L3T 188-17 L3T 188-14 L4 193-14 L4A 194-45 L4B 195-3	16-46 186-33 16-45 187-7 187-7 186-52# 187-1 186-16# 189-45# 190-6# 190-1 188-30 190-27# 190-27# 189-14# 189-28# 194-40# 194-47# 195-5	87-20 .213-24 .162-42 	99-15 217-17 197-8* 	217-18	217-32	232-31 234-16	232-43	232-55					
L25EN 9-32# L25MEM	16-46 186-33 16-45 187-7 187-7 186-52# 187-1 186-16# 186-16# 189-45# 190-6# 190-15 190-27# 189-14# 189-14# 189-28# 189-28# 194-40# 194-40# 194-23#	87-20 .213-24 .162-42 	99-15 217-17 197-8* 	217-18	217-32	232-31 234-16	232-43	232-55					
L25EN 9-32# L25MEM	16-46 186-33 16-45 187-7 187-7 186-52# 187-1 186-16# 189-45# 190-6# 190-1 190-27# 190-27# 190-35# 189-24# 189-24# 194-40# 194-40# 194-47# 195-5 174-23# 198-9#	87-20 .213-24 .162-42 	99-15 217-17 197-8* 	217-18	217-32	232-31 234-16	232-43	232-55					
L25EN 9-32# L25MEM	16-46 186-33 16-45 187-7 187-7 186-52# 187-1 186-24 187-1 186-16# 189-45# 190-6# 190-1 190-27# 190-35# 190-35# 190-35# 194-40# 194-40# 194-47# 195-5 194-23# 198-30#	87-20 .213-24 .162-42 	99-15 217-17 197-8* 	217-18	217-32	232-31 234-16	232-43	232-55					
L25EN 9-32# L25MEM .4-12# L25ON 10-22# L2A .184-37# L2B .184-49. L2D .184-55 L2T .185-18 L2TL .185-17 L3 .188-14 L3A .190-3 L3AT .188-25 L3B .190-13 L3C .190-25 L3D .190-33 L3T .188-17 L3T .188-17 L3T .188-17 L4T .193-14 L4 .194-45 L4B .195-3 L4T .193-17 L4TL .193-17 L4TL .193-17 L4TL .193-17 L4TL .193-17 L4TL .193-17 L4TL .193-17 L4TL .193-17 L4TL .193-17 L5T .194-14 L5T .194-14	16-46 186-33 16-45 187-7 187-7 186-52# 187-1 186-16# 189-45# 190-6# 190-1 188-30 190-27# 189-14# 189-28# 194-40# 194-40# 194-23# 194-23# 194-23# 194-23# 194-23# 197-44# 200-28#	87-20 .213-24 .162-42 	99-15 217-17 197-8* 	217-18	217-32	232-31 234-16	232-43	232-55					

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE S-12
CROSS REFERENCE TABLE (CREF. V01-05.)
L6T 199-17 199-54# 200-11
L&TL 199-16 200-11# L7 202-14 203-43#
L7C1 204-10 204-15# L7C2 204-12 204-14 204-16 204-18#
L7D 204-24 204-26# L7T 202-17 203-12# 203-26 L7TL 202-16 203-26#
L8 206-14 208-51#
LSA 209-3 209-5# LST 206-17 207-49# 208-9 LST2 206-36 208-19# 208-34
L8TL 206-16 208-9# L8TL2 206-35 208-34#
LCIF 27-16# 56-19 LF 2-7# 19-55 230-31 241-33 241-33
LI1 27-17 27-21# LINADH 8-16# 22-21 22-40* 44-25* 115-20
LINADL 8-17# 22-20 22-39* <b>44</b> -22* 115-19 
LKBR 32-15 32-17# LKUP1 179-20 179-24 179-27# LN1B1 184-20 184-32#
LN1B1 184-20 184-32# LN1B2 184-22 184-33# LN3B 191-2 191-51#
LN3B1 191-5 191-52# LN3B10 191-32 192-4#
LN3B11 191-35 192-5# LN3B12 191-38 192-6#
LN3B13 191-41 192-7# LN3B14 190-5 191-11 191-17 191-23 192-8#
LN3B3 191-8 191-53# LN3B5 191-14 191-54#
LN3B7 191-20 192-1# LN3B8 191-26 192-2# LN3B9 191-29 192-3#
LN369 191-29 192-3# LN3C1 190-12 192-9# LN3C2 190-14 192-10# •
LN3C3 190-16 192-11# LN3C4 190-18 192-12#
LN3E1 190-24 192-13# LN3E2 190-26 192-14#
LN3G1 190-32 192-15# LN3G2 190-34 192-16#
LN4B1 194-44 195-29# LN4B2 194-46 195-30# LN4D1 195-2 195-31#
LN4D2 195-4 195-32# LN4D3 195-6 195-33#
LN4D4 195-8 195-34# LN7D1 204-11 204-39#
LN7D2 204-13 204-40# LN7D3 204-15 204-41#
LN7D4 204-17 205-1# LN7F1 204-23 205-2#
LN7F2 204-25 205-3# LN8B1 209-2 209-19#

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE S-13 CROSS REFERENCE TABLE (CREF V01-05 ) LN8B2 209-4 209-20# 218-23# LNOFF 218-15 218-17 218-24# LNON 188-18 193-18 196-18 199-18 202-18 206-18 LOOKUP 179-19# 182-22 185-19 46-21 49-26 161-23 171-2 LORDER 11-27# 44-15 45-16 16-49 MAIN ...16-19# 2-20# 16-45 16-46 124-51 132-16 132-20 87-36 118-47 119-15 120-16 124-23 MAXCHA 22-31 24-24 52-53 87-28 49-22 87-19 99-17 99-38 99-41 113-15 115-41 116-17 116-46 121-20 MAXLIN 2-18# 24 - 2749-17 128-29 162-43 212-16 126-32 126-33 142-15 161-40 125-31 125-34 122-19 124-31 124-41 124-43 125-19 213-20 229-17 230-36 242-22 228-21 231-37 MLATCH 150-30 6-7# MN1 16-21 16-23 16-26# MN1A 16-28 16-30# MN1B 16-31# 16-29 16-36# MNIC 16-34 16-36 MN1D 16-38# ..... 16-33 16-40# 16-30 16-31 16-27 MN2 16-26 MN3 16-40 16-42 16-45# ..... MODE 10-38# 98-13* 98-31* 193-30 193-38* 193-56* 195-20 MONITO 10-29# 17-41 234-26 235-3# MSGCL MSGIC 234-36 235-13# 234-31 235-8# MSGOL 242-19 NKC 78-37 100-34# 69-37 69-38 242-18 NORM 17-29# NRMO 17-34# 17-33 ..... 17-35 NRM1 17-34 17-37# ...2-3# NULL 42-4 42-34 42-37 42-40 42-43 124-21 124-49 130-15 171-7 212-17 213-26 NUMCHA 2-21# 44-19 45-20 112-37 115-16 116-51 NUMLIN 113-19 128-28 161-24 166-26 170-48 170-50 170-55 2-19# NVR1 239-17# 239-19 NVRCHK 160-14 211-13 NVRSUM 11-4# 160-15 211-14* 29-30 29-29 242-10 242-11 OBUF 6-19# 6-20 .28-29... 29-20 29-29* 33-44 29-28 OFETCH 8-11# 29-16 29-21 29-25* 9-30# 16-27 174-46* 177-25 177-32* 231-13 OKTRAN OLEN 6-20# 28-23 10-9# ONLINE 16-26 OSTORE 8-10# 28-22 PBS 19-42 117-36# ..... PBS1 117-39 117-42# PCIF 25-15# 28-16 28-19 36-29 174-40 ..... 30-30 56-17 100-16 167-55 PCKB 30-13# 30-21 PCL1A 111-14 111-15# PCL1B 111-18 111-19# 111-23#... .PCL18...111-22 111-23 111-25# PCL2 111-15111-19 94-31 137-31 154-19 POLK ... ...68-38 111-13# 37-47 PCOF 28-15# 33-14 36-27 37-23 .....PQR.... 47-14 47-35 48-13<del>*</del> .22:33. PCRLF 20-2 47-14# 61-28 132-16# 133-19 PDC PDC1 132-35# 132-56 PDC2 132-33 132-57# PDL 61-27 63-47 128-15# 129-19 PDL1 128-23 128-27#

Z29ROM; MICROBENCH 8051	CONCC ACCEMBLE		74_FER_83 15:30	:06 PAGE S-1	 Д				
CROSS REFERENCE TABLE ()									
PDL2 128-27 128-32#									
PDLA 128-20 128-21# PFIELD 11-21# 23-14	55-22* 55-:	 29* 55-3 <b>4</b> *	92-36* 112-	14 112-15*	112-17*	113-44* 1	13-46* 168	3-39* 172-25	 5 173-1
	226-29* 230-								
PI2 25-27 25-30# PI3 25-33 25-36#	26-35 29-3	 32							
PIC 18-41 130-15# PIC1 130-27# 130-43							••••		
PIL 61-26 63-44 PILO 126-27# 126-34	126-18# 127-	19						••••••	
PIL1 126-25 126-30# PIL2 126-30 126-36#									
PILA 126-22 126-23# PILRET 126-28 126-40#	128-25 128-								
PILRT1 126-23 126-32 PIR 25-24 25-37#	126-41# 128-	21 128-28	,						
PK0 30-18 30-20# PK1 30-15 30-23#									
PK1A 30-27 30-32# PK2 30-36# 30-42									
PK3 30-36 30-45# PK4 30-47# 30-48									
PK5 30-47 30-51# PKDLY 30-37 30-38 PKL 30-34# 30-51	30-39 30-	30-41	30-54#						
PKL 30-34# 30-51 PKR 30-43 30-49 PLF 22-34 47-15	30-53# 47-34 47-:		65-40 128-						
PLF1 49-17 49-19# PLF2 49-20 49-22#									
PLF3 49-21 49-24# PLFCR 19-56 47-34#									
PLFLP 49-38# 49-42 PLFRET 49-22 49-47#		• • • • • • • • • • • • • • • • • • • •							
PMADRH 8-25# 83-20* PMADRL 8-26# 83-19*	83-4383-! -83-42 83-!		**************************************	Ż9¥ 28∗		• • • • • • • • • • • • • • • • • • • •			
PMBUF 6-22# 83-19 PMNUM 8-27# 83-21*	83-20 83-7 83-36 83-7		85-18 83-54* 84-	1 84-3*	85-22	85-30*			
PMVALU 8-28# 83-23* P01 28-15 28-18#	83-31* 83-:	83-35*		57*					
P02 28-18 28-21# P03 28-23 28-28#									
FORTF 9-7# 16-28 PRINT 158-16 230-15#	160-6****177-								
PRLF 46-16# 61-23 PRNT1 230-28# 230-36	65-34 126-	38							
PRNT2 230-45 230-48# PRNT3 230-51 230-54#	a a ay ay ya a ay ay a a a a a a ay ay a	koa ninnoaakoa nishkii							
PRNTF 9-53# 174-30 PROTEC 9-21#		39 230-15*	230-48*						
PRTF2 10-31# 160-5 PRTYEN 10-3# 177-17		39* 188-47*				· · · · · · · · · · · · · · · · · · ·			
PRTYEV 10-5# 177-21 PRTYST 10-4# 177-19 PSD 66-18 73-17	188-41* 188-4 188-40* 188-4 74-17 83-	13 188-49*	400 45						
PSD1 83-37 83-39# PSD2 83-49 83-52#									

729ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE S-15	
CROSS REFERENCE TABLE (CREF VOI-05 )	
PSDB 83-17 83-25# PSDB1 83-25 83-26#	
PSDB2 83-27 83-28# PSDE 83-26 83-28 83-42#	
PSDF 9-38# 58-14* 73-14* 74-14* 83-17 83-18*	
PSOF 37-20 136-14# 136-23 137-25 138-16 138-42 139-18	46-39 49-28 49-31 49-39 49-40
49-44 50-13 50-17 50-18 51-22 51-23 51-33 52-24 52-30	52-50 130-24 130-36 130-39 132-24
132-48 132-54 159-16 159-22 159-34 159-46 159-48 160-27 160-29 187-8 211-16 211-20 211-27 217-19 217-25 217-26 217-30 217-31	161-23 161-26 161-27 186-35 186-37 239-14 239-17 239-18
PTR1A 7-8# PTR2 7-9# 46-25 46-26 46-33 46-36 49-29 49-30 49-38 49-41	130-25 130-33 130-40 132-29 132-51
132-55 185-14 185-25 185-26 185-27 185-33 185-34 185-35 185-48 PTR2A 7-10# 186-49 187-1	
PU1 22-31 22-37# PU2 22-32 22-35 22-42#	
PUTCHR18-4422-14#	
R2_DCL 14-34# 16-47 158-21	
R2_FIL 12-35# 112-39 114-20 115-21 116-53 156-34 R2_ICU 13-14# 101-33 157-6	
R2_IKB 13-19# 39-21 157-12 R2_IN 12-11# 16-8 156-10	
R2_IN2 12-17# 96-19 156-16 R2_IRQ 12-40# 15-34 156-40	
R2_KY1 13-36# 31-15 157-30	
R2_PRN 14-29# 38-17 61-46 95-17 158-15	
R2_SER. 12-46# 15-43. 156-46	
R2_STA 12-23# 15-49 156-22 R2_STO 12-29# 15-50 156-28	
R2_T2514-22#61-43159-9	
R2_TL 14-10# 61-44 157-54 R2_TP 14-16# 60-17 158-3	
R2_XMT 13-4# 16-38 156-52 R2_XON 13-9# 26-21 156-57	
RAMO 238-18# 238-40	
RAM2238-38238-42#	
RAMCHK 161-35 207-19 238-15# RAMCQN_159-17 242-6#	
RAMERR 238-30 238-33 238- <b>44#</b> RAMP 62-15 96-19#	
REPF 10-19# 167-50 199-36* 201-2 RGL 56-19# 56-23	
RGR 56-20 56-21 56-24# RING 20-5 56-16# 207-29 228-29 230-53 240-27	
RL1 46-17 46-45#	
RL1A 46-45 46-47# RLL 46-33# 46-37	

		<u>.</u> ,										
Z29ROM; MICROBENCH 8051 CROSS REFERENCE TABLE (C												
RLR 46-42# 46-48	•••••			• • • • • • • • • • • • • • • • • • • •								
RMC1 237-19# 237-24 RMC2 237-27 237-29#	237::25									• • • • • • • • • • • • • • • • • • • •		
RMS 62-14 80-14#												
RMS1 80-20 80-21# RMSRET 80-19 80-21	80-27# 81-	14 81-18										
RMST 80-24 81-16 RMSTL 80-23 81-15	81-30# 82-	2										
ROMCHK 161-34 207-18	82-2# 23 <b>7-14#</b>	• • • • • • • • • • • • • • • • • • • •										
RT1 98-32# 99-15 RTS 11-42# 178-18*	178-45*								• • • • • • • • • • • • • • • • • • • •			
RUBOUT 2-13# 17-51 RVVA 9-15# 17-42*	214-36 220-								• • • • • • • • • • • • • • • • • • • •			
SAV2 211-20# 211-28	17-45* 143-	23* 145-13*	145-31*	222-8								
SB1 150-17 150-20# SB2 150-21 150-26#												
SB3 150-24 150-28#	• • • • • • • • • • • • • • • • • • • •					• • • • • • • • • • • • • • • • • • • •						
SBC 78-43 101-13# SBLR 68-29 150-16#										• • • • • • • • • • • • • • • • • • • •		
SBR 95-40 134-22 SBRO 134-48 134-49#	134-42# 188-	32										
SBRET 134-18 134-20	134-44 134-	49 134-51#										
SBRT 150-22 150-33# SC1 164-30 164-32#												
SC2 164-38 164-40# SCH 19-34 61-22										• • • • • • • • • • • • • • • • • • • •		
SCH 19-34 61-22 SCLRL 210-18 213-19#	63-14 99- 234-21 240-	42113-27 30	11/-13#	12373/	• • • • • • • • • • • • • • • • • • • •				• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •	
SCLRLR 182-16 196-24 SCP 61-56 65-22	196-37 206- 141-13# 141-		213-17#						• • • • • • • • • • • • • • • • • • • •			
SCRN1 5-4# 164-26												
SCRN2A 5-5# 164-29 SCRN2B 5-6# 164-31												
SCRN3 5-7# 164-34 SCRN4A 5-8# 164-37												
SCRN4B5-9#164-39				• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •			• • • • • • • •
SCRNSA 10-30# 175-51 SD1 169-25 169-26	_197-20*198-; _169-29#	21										
SERIAL 156-47 174-13# SERIN 174-18 174-22#										• • • • • • • • • • • • • • • • • • • •		
SEROUT 174-19 174-45#												
SETCF 157-7 162-39 SETCRT 163-23 164-13#	165-14# 202-! 165-14	52 203-2										
SETDIS 66-16 69-17# SETLCK 11-40# 182-13	73-15 74-	15 77-14	80-14	95-13	124-13	134-13	138-37	140-13	143-14	147-26	147-35	
SETNOR 17-23 58-19	58-3860-	14 66-20	69-37#	73-19	74-19	77-16	80-16	95-15	124-38	134-15	138-39	• • • • • • • •
140-15 143-16 SETTAB 51-15# 54-15	147-28 147-: 185-49	37										
SETUP 157-19 182-13#								· · · · · · · · · · · · · · · · · · ·		• • • • • • • • • • • • • • • • • • • •		
SETUPO 182-13 182-16# SETUP1 182-18# 182-30	182-42 182-4	45 182-48										
SETUPZ 182-51 182-54 SETUP3 182-57 183-3	185-14# 188-14# 188-:		188-46	188-48	188-50							
SETUP4 183-6 183-9	193-14# 193-	25 193-42	193-49	193-51	193-57				• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •		
SETUP5 183-12 183-15 SETUP6 183-18 183-21	196-14# 196-3 199-14# 199-3		196-46 199-38	197-3 199-45	197-9	197-15	197-21			• • • • • • • • • • • • • • • • • • •		
SETUP7 183-24 183-27 185-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 1163-33 11	202-14# 202-2 206-14# 207-4	25 202-31	202-53 207-28	203-3 208-26								

229ROM; MYCROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE S-17 CROSS REFERENCE TABLE (CREF V01-05 ) 210-14#11211-35111 SETUPR 183-36 208-29 211-13# SETUPS 183-39 208-32 SF1 172-35 172-40# SF2 172-40 172-43# 172-38 SF3 173-2 173-4# SF4 173-9 173-12# 173-16# '\$F5' 173-10 173-13 SF6 173-21# 173-11 173-14 SFA 172-28# 172-23 SFB 172-21 172-26 173-1# 181-20 182-19 185-16 188-15 193-15 196-15 199-15 202-15 206-15 206-34 214-24 SFCKB1 181-18# 181-19 SFCKB2 181-17 181-21# SFIL1 212-17# 212-21 SFIL2 212-18# 212-20 SFILL 206-32 207-3 SFL 173-6# 173-22 37-15 154-23 SHIFTE 9-42# 33-18* SIO 174-27 174-30# 174-30 174-32# SII SITA 174-33 174-36# 174-32 174-34 SI2 Sis 174-37 174-39# 213-29 SLP 213-27# 77-14# SMS 62-13 SMS1 77-20 77-21# 77-19 78-14 77-21 77-27# SMSRET SMST 77-24 78-16 78-31# 77-23 SMSTL 78-15 79-3# SOFTI1 162-42 162-46# 162-15# SOFTIN 156-17 SOR 136-16 136-25# SPO 24-20 24-23# SP1 24 - 2724-31# 24-24 24-34# SP2 SP3 24-32 24-35# 22-42 24-19# 24-36 SPF 22-18 18-40 SRET 210-14 210-15 210-17# 174-39 174-31 174-43 SRT 174-20 174-28 174-41 174-48# SRT1 174-53 174-56# 21-21# STI 21-39 21-37# ST2 21-24 33-30 STAB 18-7 21-19# 86-20 180-32 206-37 STACK 11-32# 16-7 START 15-9 16-7# STARTD 156-23 169-17# 51-32 51-25# STB1 STB2 51 - 2651-29# 72-19 75-19 76-19 STCALL 70-21# 71-19 78-19 STFILL 156-35 172-18# 58-20 77-25 140-18 STUMP 66-23 70-18# 73-22 74-22 80-25 STOPDM 156-29 170-17# STP1A 182-19# 182-23 182-28# 183-53 STP2A 185-15# 185-28 185-20 STP2B 185-16# STP2C 185-25# 186-5

Z29ROM; MICROB	ENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE S-18
CROSS REFERENC	E_TABLE_(CREF_V01-05_)
STP2C1 185-26	185-28#
STP2D 185-33# STP2D1 185-34	186-8 185-36 <b>#</b>
STP2E 185-41#	186-11
#STP2F 185-47 STP2F1 185-42	105 404
STP3A 188-15#	***** *********************************
STP3B 188-24# STP3B1 188-28	
STP30 188-38#	188-30 188-32# 189-20
STP301 188-38	188-43#
STP3C2 188-43 STP3C3 188-44	188-47# 188-49#
STP3D 188-55#	
STP3E 189-4#	189-26
STP4A 193-15# STP4B 193-24#	
STP4C 193-30#	194–15
STP4C1 193-35 STP4C2 193-40	193-37# 193-42#
STP4D 193-47#	
STP4D1 193-47 STP4E 193-56#	193-50# 
STP5A 196-15#	
∵STP5B 196-24#	•
STP5B1 196-32 STP5B2 196-34	196-34# 196-35#
STP5B3 196-45	196-46#
STP50 197-8# STP5D 197-14#	197–36
"STP5E" 197-20#	
STP6A 199-15#	
**************************************	200-2
```\$TP6D```199-36#	200-6
#STP6E 199-43 #STP7A'''202-15	
STP7B 202-24#	203-15
**************************************	¹¹ 203-18
STP7D 202-36# 	203-21 202-41#
STP7D2 202-36	202-45#
``STP7D3`202∸45`` STP7D4 202-39	202-49# 202-43 202-47 202-52#
"STP7E" 203-1#"	203-24
STP8A 206-15# STP8B 206-24#	
STP8B 206-24# STP8B1 206-31#	
**************************************	208-22
STP8C 206-48# STP8D 207-9#	208-5 207-52
STP8E 207-17#	
STPSE0 207-20#	207 07
STP8E1 207-32# STRET 66-19	207-37 70-19 70-24 #
SUC 81-42	101–31#
SUC1 101-14	101-33#
•••••	

729ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE S-19 CROSS REFERENCE TABLE (CREF V01-05) T\$X01 226-14 226-26 SXMTC 157-49 226-14# ···227-15# 157-55 226-18 SXMTL 178-47# 178-49 SXON1 178-44 178-49# ˈˈsxrˈ 178-14 178-41 52-23 186-35 TAB 19-45 52-15# 53-23 TABTAB 10-48# 50-13 51-21 52-32 52-47# TB1 52-42# TB21 52-40 52-35 TB3 52-41#TB4 152-37 52-39# TB5 52-46 52-50# TB6 52-31 52733# 52-30# 52-51 TBL1 TBL'2 52-35# 52-44 TBR 52-41 52-55# 162-37* 175-41* 175-42 175-47* 197-2* ·····TCNT 11-14# 1762-15 35-19 35-37 35-40 35-51 36-19 45-15 45-22 52-16 52-36 33-45 36-16 TEMP 7-17# 33-4260-28. 84-2 94-34 94-39 118-41 120-17 130-31 132-43 159-41 159-45 60-32 83-44 83-55 84-1 187-6 193-31 173-4 186-40 186-48 186-57 193-37 238-27 170-24 170-25 170-35 173-10 173-12 186-34 TEMPA 7-18# 132-55 193-32* 238-30 46-39 49-31* 49-44 118-44 120-20 130-40 173-13 46-27* TERMIN 168-15# 193-39 176-7 176-9* 111-16* 162-18 162-34* 176-6* 196-36* 232-25 THOUR 11-17# 162-35* 176-2* 176-3 176-5* 196-47* 232-39 TMIN 11-16# 111-20* 162-17 177-23# TRA 177-17 177-19 177-21 TRANS 177-16# 178-51 178-23 177-14# 219-22 219-26 TRANSM 156-53 177-14 TROK 177-25 177-27# 177-27 177-29# TROK1 177-29 TROK2 177-28 177-31# 177-25# TRW 177-26 126-22 126-37* 126-42* 128-16 TSCROL 8-22# 46-17 46-29 49-33 121-17 125-21* 125-27 126-19 128-20 128-34* 154-26 .JeU 11-15# UNDEF /* 175-48* 175-49 162-36* 176-1* 197-1* 232-21 232-24 232-51 111-24* 162-16 61-49 61-50 61-51 61-32 61-33 61-34 61-35 61-36 61-37 61-38 61-47 61-30 61-31 61-52 62-37# 61-53 61-54 62-2 62-17 62-19 62-20 222-29 143-29* UNDLNA 9-16# 144-31* USCP 61-57 65-25 142-13# 142-36 USCP1 142-15 142-17# 142-18# USCP2 142-16 16-37* 154-16 178-49 9-28# UXOFFS 16-35* 11-39# VSP. 169-27* 170-52* 175-56* 174-23* 175-55* VSPF 161-42* 169-26 9-54# 31-22* WEOL 62 - 1175-41 97-15# WLATCH 159-30 159-32 160-7 175-39 211-31 211-33 231-36# 23-31 30-46 WORK 7-15# 21-19 21-33 21 - 4123 - 1923-29 49-35 51-17 51-31 52-18 52-43 52-48 52-49 52-53 52-55 113-19 113-25 46-31 46-37 49-42 225-23 225-24 130-28 215-11 215-12 225-26 225-28 228-20 228-21 132-46 214 - 20214 - 29215-8 230-36 237-14 237-21 237-22 238-16 238-21 238-28 238-38 238-39 WORK 1 7-19# 186-39 186-52 186-54 214-17 214-42 214-45 215-17 215-19 7-20# WORK1A 172-19 WORK2 7-21# 172-21 173-16 186-46 186-50 186-56 187-2 187-3 196-32 214-19 214-40 214-43 214-53 241-12 241-17 241-20 241-25 241-28 215-20 240-44 240-47 241-1 241-4 241-9 7-22# WORK2A 130-39 132-54 238-33 WORKA 7-16# 21-24 23 - 3252-37 XACR 72-45 81-57 107-32#

81-54 108-32#

..... Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE S-20CROSS REFERENCE TABLE (CREF V01-05.) 63-53 XATR 91-14 143-49# ...XBLINA...86-8# .87-32.. ...117-16....117-43....118-48....119-19....120-24....121-24.....122-27....124-36....124-53....126-44....142-18 XERM 72-42 110-34#XGATM....72-33. 109-34# XGM 91-32 61-21 146-31# XHSM 81-39 61-42 102-32# 193-48 XHSM1 102 - 15102-34# XICM 61-29 72-39 105-31# XKAM 61-12 65-31 81-51 104-31# XKSM 62-10 81-48 103-31# 220-21 XLO. 220-23 220-28# XLOA 220-44 220-45# 220-45 220-53# XL1 222-29 XL10 222-35# XL10A 222-33 222-37# XL11 222-31 222-35 XL12 222-41 222-47# XL12A 222-45 222-49# 222-43 XL13 222-47 222-53# 222-53 XL14 223-2# XL14A 222-57 223-4# XL15 222-55 223-2 XL16 `223-8 223-22# 223-35**#** XL16A 223-26 223-31# 7223-13 223-25 XL16B XL17 223-10 223-17 223-20 223-22 223-29 223-33 223-39**#** XL18 223-41 224-9# XL18B 224-12 224-16# 224-16 XL180 224-20# XL18D 224-20 224-24# XLIBE 224-24 224-28# 220-53 XL1A 220-56# 221-1# XLZ 220-51 220-55 224-31# XL20 223-39 224-5 XT3. 220-46 221-5¥ XL3A 221-16 221-21# XL3B 221-15 221-25# XL3C 221-14 221-31# XL4 [221-9] 221-35# 221-50 XL4A 221-40 221-45# XL4AA 221-41# 221-54 XL4B 221-39 221-50#XL'40' 221-38 222-1# XL4SET 221-23 221-47# 221-29 221-33 221-35 221-43 221-48 222-6# XL7 221-11 221-19 XLS 222-8 222-17# [222-26] XESA. 1222-12 222-25#1 XL9 222-9 222-15 222-17 222-23 222-29# XLATCH 9-3# 150-29 231-36 XLON1 218-16 218-18# XLONOF 195-15 195-21 201-3 201-9 203-49 204-5 218-15# 198-10 198-16 198-22 200-34 200-40 XML2 225-21# 225-24 XMP1 228-18# 228-21 XMP2 228-23# 229-19 XMP3 227-16 229-15 228-26# XMT25 158-10 226-22 229-15# XMTC'' 220-18# 225-21 226-28

Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE S-21 CROSS REFERENCE TABLE (CREF V01-05)

XMTL	225-18#	227-15	229-18				• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •					
XMTL1	225-19#	228-18	. 230-28											
XMTP	158-4	226-16	228-15#								000 04			000 0 7
XMTS	.219-16#.		221-21	221-26	. 221-31 22 4- 17	221-41	221-45 224-25	. 221-51 22 4 -28	. 222-1 . 228-26	222-13 230-30	222-21 240-36	223-14 240-45	223-18 241-2	223-27 241-10
	223-31	223-45												
YMTS1	241-18 219-19#	241-26	4847.94				• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •				
	219-21	219-26#												
	178-17	178-22#			,									
XMTXON	.178-50#.													
XOFF	2-10#	16-34			178-22									
XOFFRC		16-31		174-38*		231-16*.								
XOFFSE		178-14 16-36	178-15* 154-17	178-41 174-37	178-42* 178-50									
	2-9#	‡9739 100-36	151-15	151-34	152-18	• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •				
XPOS		22-30	22-37*	23-20	24-23	24-25*	24-34*	44-21	48-13*	52-15	52-55*	53-15	53-18*	53-20
	53-26	53-29*	54-14	55-14	55-20	55-24	55-31	55-42*	87-31*	94-26	114-17	115-18	117-14*	117-38
	.117-42*.		118-47*	112-14	119-18*		120-23*		126-40*	130-17	130-20	132-18	137-15	137-29
	141-13	142-13*	166-20	220-28	225-19	225-20*	225-22*	225-23	225-25*					
XR2_DI.			1.05-1.4	105-32	152-19							• • • • • • • • • • • • • • • • • • • •		
XRVM	62-6	145-31# 198-6	198-12	198-18	200-30	200-26	200-42	201-5	203-45	204-1	204-7	204-20	204-28	206-27
X\$T	49979 207-2 4	#79T9 208-53	209-7	214-46	216-17#	20090						+87+8		
XSTAB		71-17		75-17		78-17	81-17	86-20#	90-18					
	179-19	179-23	180-32#	188-27	190-2									
XSTAT				209-5	215-18	216-29	217-17#	218-18	234-28	234-33	234-38	240-41	240-49	241-6
	241-14	241-22	241-30											
			179.7	1.90-19	190-27	189535	1.94.74./	19579						
XSTL	217-21#													
XŞTR XSTX	. AIATAT 180-8#	184-17	184-25	189-47	190-9	190-21	190-29	194-41	194-49	195-11	195-17	196-27	196-40	• • • • • • • • • • • • • • • • • • • •
XXPQS														
XYPOS	11-7#	141-14*	142-14											
YPQ\$				24-26			44-1.6		46-41	46-42*	49-16		49-47*	,55-15
	55-21	55-25	55-32	55-41*	87-23*	94-21	99-40	113-14	114-15	114-38	115-40		116-15	116-44
		1.18-41	118-43 128-18		120-19		121-23*. 132-21	1447.19 137-20	122-26*. 137-26	124-35*	140516 142-17*	126-21 166-23	126-26 220-29	126-37 228-16
	126-43*	128-15					. 230-25							
ZAR	138-40	139-16	143-19		143-32#		1777 1777							
ZATR		143-14#												
ZATRO	143-20	143-21#												
			3718	57.–32	195-5									
	134-19	134-20#												
ZDSESÇ ZDSSBR		57±30 134=13#	69:14#.						• • • • • • • • • • • • • • • • • • • •					
	62-7 60-38	61-10#												
ZE1A	60-16	60-18#			• • • • • • • • • • • • • • • • • • • •									
ZE1B														
ZE1C	60-20	60-23#												
ZE2	58-40		60-36#.											
ZTAB	60-21	140-13#	140.40											
ZTABT	140-17 . 140-16	140-40#	49.749											• • • • • • • • • • • • • • • • • • • •
LIMBIL	. 140-16	140-40#												

	Z29ROM; MICROBENCH 8051 CROSS ASSEMBLER (V1)-218 24-FEB-83 15:30:06 PAGE M-1
	CROSS REFERENCE TABLE (CREF. V01-05.)
	\$\$INIT 0-0#
	·

1.0	
	·
	•
	······································
	· · · · · · · · · · · · · · · · · · ·

C